(1)

A Diffusion Link Laser Programmable Read Only Memory and Automated Programming Station

THESIS

John J. Tillie
Captain, USAF

AFIT/GE/ENG/88D-56

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/88D-56

A Diffusion Link Laser Programmable Read Only Memory and Automated Programming Station

THESIS

John J. Tillie
Captain, USAF

AFIT/GE/ENG/88D-56

Approved for public release; distribution unlimited

AFIT/GE/ENG/88D-56

A Diffusion Link Laser Programmable Read Only Memory and Automated Programming Station

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

John J. Tillie, B.S.E.E.
Captain, USAF

December, 1988

Acknowledgments

I am grateful to the members of the VLSI reseach group without whose help, this research effort would not have been completed: Captain Erik Fretheim, Captain Keith Jones, and Captain John Comtois. I am also thankful to all the other members of the VLSI and Devices research group for their assistance, encouragement and comradery. I would like to acknowledge the members of my thesis committee Major Joseph DeGroat and Captain Brian Donlan with a special thanks to my thesis advisor Captain Richard Linderman for his guidance and encouragement.

John J. Tillie

Table of Contents

List of Figures

List of Tables

Abstract

Laser programmable read only memory circuits expand the applications of VLSI circuits. This thesis considers a laser programmable diffusion link as vehicle for implementing a high speed static memory circuit.

This ~~research effort~~ thesis involves designing a laser programmable ROM and designing, assembling, and demonstrating an automated laser programming station. This system includes a laser and optics, a stepper motor controlled X-Y translation stage, and a camera with pattern recognition firmware.

The capabilities and limitations of this automated system are examined. The layout of the LPROM circuit illustrates design considerations for integrating a laser programmable diffusion link into a VLSI circuit. The required accuracy of the programming station is obtained at the expense of the time necessary to execute the pattern recognition firmware.

Keywords: Systems engineering, fabrication, (KP)

A Diffusion Link Laser Programmable
Read Only Memory and
Automated Programming Station

I. Introduction

1.1 Background

Generating full custom very large scale integrated (VLSI) circuits, such as the application specific math processor designed at the Air Force Institute of Technology by Capt D. Gallagher in December 1987, may require several weeks or months to fabricate and verify [Gal87]. The result is a very powerful circuit limited to the functions defined in the memory of its control section. The flexibility of such a circuit could be increased if its components and memory were reconfigurable or programmable after fabrication. Lincoln Laboratories in Massachusetts has been successfully using lasers to reconfigure wafer scale integrated circuits to increase their yield [Lin85]. In December 1987, Capt C. Spanburg designed a laser programming system and demonstrated that a laser could be used to customize Metal Oxide Semiconductor Implementation Service (MOSIS) fabricated circuits on a small scale [Spa87]. Thus it seems reasonable to believe that a VLSI circuit could be designed with a laser programmable memory that could be further customized to better meet the end user's needs.

Specifically, the flexibility and number of applications of a given VLSI circuit can be increased by permitting the end user of the circuit to insert his own microcode into the on-chip memory in addition to any preexisting code. Integration of the microcode into the ROM prior to fabrication would require the end user to wait for MOSIS to fabricate the circuit. This can take up to twelve weeks. A high speed laser programmable ROM (LPROM) could be integrated in the memory of the control section of a large VLSI circuit. The circuit could then be fabricated and stored until needed, at which point the user's microcode would be "burned" into the LPROM portion of the on-chip memory. Turn around time on a circuit using this process would be measured in hours.

1.2 Problem Statement

A laser programmable read only memory (LPROM) capable of being integrated into large VLSI circuits needs to be developed to increase the number of applications of a given circuit through the customization of its own internal memory. To ensure compatibility with other design efforts, this LPROM and it associated programming apparatus must integrate with, and make use of, existing CAD tools. The VLSI design will conform to design rules established both by MOSIS and this research effort.

1.3 Approach

This section outlines the basic philosophy of how the above problem will be solved. This thesis effort can be divided into two main sections; to generate a VLSI design for an LPROM and assemble an automated laser programming station.

1.3.1 **Generate a VLSI Design for the LPROM.** The XROM provides a proven architecture upon which to base the LPROM circuit. Possible solutions to the LPROM design problem will involve inserting a linkable device between a memory location transistor and a ground line. The LPROM will have transistors with floating drains fabricated into all its memory locations. The LPROM will consequently be initialized to all *0's*. The laser will be used to tie the transistor's drain to a ground line and program that particular memory location to a *1*.

Two linking techniques developed by Lincoln Laboratories in the customization of their wafer scale integrated circuits are the fusable link and fusable via. Both of these devices represent potential solutions and are discussed in greater detail in Chapter 2. Due to its compatibility with MOSIS design rules, the fusable link holds greater promise than the fusable via for use in the LPROM.

1.3.2 **Assemble the Laser Programming Station.** The laser programming station consists of three main parts; the laser, the optics box and the X-Y translation table. Previous research indicates that a 70 to 80 μsec pulse of a 510 nm wavelength laser beam at a power of 3.5 watts is optimum for linking [Lin88]. This can ideally be generated by

either a Spectra Physics 2020 or 2016 argon-ion laser. Computer controlled stepper motors tied to an X-Y translation table position the LPROM beneath the laser. Accuracy requirements are ensured by the use of pattern recognition software developed by Cpt. E. Fretheim. This software is integrated into the control software and receives its input through a video camera mounted on the optics box. A two stage shutter, designed by Lincoln Labs, controls the duration of the laser pulse into a Florod optics box. This box is designed to collimate, shape, guide, and focus the laser beam down to the LPROM circuit. This device also contains a spotting light which, when projected through the laser's path, can be used to test the positioning accuracy of the proposed design.

1.3.3 **Develop Automation Software.** Software will have to developed to automate the programming station. This software will be written in the "C" programming language and will accept an input data file of compiled microcode; determine the actual size and orientation of a fabricated LPROM; maneuver the LPROM beneath the laser; and, finally, control the laser shutter to program the desired memory locations. Pattern recognition software will be used in determining the actual size and orientation the LPROM and, to cancel mechanical positioning errors at the beginning of each row of memory locations.

1.3.4 **Test Circuits.** Upon completion of the programming station, the MOSIS fabricated test circuits will be laser programmed with a test code. The contents of each address will be examined to verify the programming process. A video tape will be made of the laser programming of the LPROM. This tape will be studied to determine the effectiveness and accuracy of the automation software.

1.3.5 **Integrate the LPROM with the XROM.** The final phase of this thesis effort will be to integrate an LPROM into a floating point application specific processor (FPASP). It will be inserted adjacent to the existing ROM and use the same address and data busses. Its exact size will be determined by the microcode requirements of the algorithms to be programmed into the FPASP.

1.4 Scope

This thesis is bounded both in the type of problems the reported solutions apply and in the scope of the solutions it presents.

The LPROM programming station is designed for functionality and will not necessarily be optimized for speed. Its hardware and software are designed to program only the LPROMs designed in this thesis effort. Any changes to the VLSI design of the LPROM will require modifications to both the hardware and software.

Similarly, the LPROM designs developed as part of this thesis are designed first to be functional. Designs will be limited to those compatible with fabrication by MOSIS. This will improve the portability of a successful design to other universities. Based upon the results of this work, follow on efforts may be able to increase the density and operating speed of the LPROMs.

1.5 Assumptions

1.5.1 **Laser Parameters.** It is assumed that the laser parameters developed by Lincoln Laboratories for programming the fusable laser link will be valid for this research effort. These parameters include the frequency, power and the pulse duration. Specifically, the center frequency for a multispectral argon-ion continuous wave laser should be 513 nanometers. The pulse length should be approximately 60 to 80 μsec and the total power output across the spectral band should be approximately 3.5 watts.

1.5.2 **LPROM design.** The VLSI design of the LPROM will be based upon the proven designs of the XROM. Due to their inherent similarities, where ever possible, XROM design cells will be used rather than design a new LPROM cell for the same task.

1.5.3 **Vendor Invariant Process.** Lincoln Laboratories has indicated that their linking techniques are vendor invariant. Thus, it is assumed that laser programming techniques validated with 2 μm test circuits in this thesis are portable to other vendors.

1.6 Limitations

1.6.1 **MOSIS Fabrication Schedules.** The available MOSIS fabrication technologies and schedule for fabrication will dictate the size of the minimum features of the test circuits. While the goal of this thesis effort is to generate an LPROM in 1.2 micron technology, the MOSIS standard *TinyChip* will be used as a test circuit. Additional LPROM test circuits are "piggy backed" on larger 1.2 μm circuits. However, there is no estimated time of arrival for these circuits. Two microns remains the smallest, regularly scheduled technology available for this circuit.

1.6.2 **Hewlett-Packard Fabrication Specifications.** Recent research by Lincoln Laboratories indicates that the Hewlett-Packard 1.2 micron feature integrated circuit fabrications use very light *n* and *p* type doping. [Lin88] Links made under these conditions may be of very high resistance, approaching 1000Ω. The 1.2 micron technology LPROM test circuits integrated into 16 point Winograd Fourier Transform (WFT) processor circuit and the test links integrated into the 17 point WFT chip can be used to verify the SPICE results which indicate the higher resistance links have a minimal effect of the operating speed of the LPROM. Unfortunately, neither of these circuits are planned to be fabricated in time to test as part of this thesis effort.

1.6.3 **Budgetary Constraints.** Due to budget constraints, many of the pieces of equipment used in this research effort may have to be adapted to perform tasks for which they may not be ideally suited. This follows from the requirement to develop an "in-house" LPROM programming capability using as many pieces of existing equipment as possible.

1.6.4 **Limited Resources.** Equipment items such as lasers, image processing boards and air suspension tables are in limited supply. Their limited availability causes scheduling conflicts.

1.7 Sequence of Presentation

Chapter 1 has provided the background and defined the problem to be solved by this thesis effort. Chapter 2 analyzes the problem through previous and related research

efforts in the field of laser programmable circuits. Chapter 3 presents the VLSI design of the complete LPROM circuit. Chapter 4 discusses the hardware and software of the laser programming station designed in this thesis effort. The results of testing the hardware and software of this research effort, including changes to the original designs proposed in chapters 3 and 4, are reported in chapter 5. Chapter 6 includes the interpretation of the results and a discussion of possible enhancements to the programming apparatus, the software, and the VLSI design of the LPROM. Future research efforts in this area will be the final topic discussed in this chapter.

II. Detailed Analysis of the Problem

2.1 Overview

The LPROM falls into the category of restructurable VLSI (RVLSI) circuits. Lasers have been one of the primary tools for performing the post fabrication modifications. This chapter presents some of the results of previous laser RVLSI research efforts in an attempt to better understand the function and construction of these circuits. After this discussion, the chapter continues with a detailed examination of the physical layout of the AFIT XROM and its components followed by a detailed, step by step description of its operation. This chapter lays a foundation for understanding how the hardware, software and VLSI design must interact to meet the requirements of the problem statement.

2.2 Previous Laser Programmable Circuit Efforts

Three research efforts with results related to the development of the AFIT LPROM are Lincoln Laboratories' wafer scale integration effort, Texas A&M's CMOS LPROM, and Capt. Craig Spanburg's laser programmable CMOS circuits and programming station designed at AFIT.

2.2.1 **Lincoln Laboratories Wafer Scale Integration efforts.** This discussion of Lincoln Laboratories efforts is limited to their use of lasers on the unique VLSI devices they developed to solve some of wafer scale integration's inherent problems.

Lincoln Laboratories initially applied laser programming to solve routing problems encountered during wafer scale integration. The laser was first used by Lincoln Laboratories to cut metallization lines. This procedure performs well for excising flawed circuits. To solve the problem of joining two nodes, first the fusable via and then the fusable link were developed. These two devices, together with the cut metallization processes, provide a means of routing around defective circuit created as a result of less than 100% yield.

2.2.1.1 **Fusable Vias.** Simply described, the fusable via consists of a specially processed *weld* point between two overlapping layers of *metal1* and *metal2*. To process

the weld and join the two nodes, a one microsecond light pulse from 1-2 watt argon laser focused to a 10 μm spot is used to first melt the *metal2* layer. A crater forms as the *metal2* flows thus exposing the amorphous silicon (a-Si) layer to the laser. As the *a-Si* melts, it combines with the molten *metal2* to form a metallic Al-Si conducting path to *metal1*. The pre-weld resistance is typically above one megaohm while the post-weld resistance is very low, on the order of one ohm. The relatively low power per square micron prevents splattering of the metals and silicon and reduces residual redeposition of this material elsewhere on the circuit [Lin85].

Follow-on experimentation has been performed at Lincoln Laboratories to attempt to further reduce the size of the via area. A report released in March of 1985 indicated that they were successful with a 50% reduction in area [Lin86]. Figure 2.1 is a photograph of their circuits demonstrating both cut metallization and fusable vias.

Unfortunately, the fabrication steps necessary to form these weld points are not available in present MOSIS fabrication technologies and are fabricated only under contract with Lincoln Labs. To realize a fusable via, the circuit would have to be partially fabricated by MOSIS and then forwarded to Lincoln Labs for the deposition of the thin passivation oxides and the amorphous silicon. This violates one of the requirements of the AFIT LPROM circuit.

2.2.1.2 **Fusable Links.** Since the development of the fusable via, Lincoln Labs has engineered a second means to join two signal lines through what they have termed a fusable link.

The fusable link consists of two like type diffusions placed back to back to each other separated by 4 microns [Lin86]. These diffusions are then connected to various metal and polysilicon lines by standard *n*-diffusion or *p*-diffusion contacts. Figure 2.2 diagrams the cross-sectional and top views of a fusable link. Figure 2.3 is a photograph of actual programmed laser links.

A 4 watt argon ion laser is mechanically shuttered to 60-80 μsec and is optically focused to deliver a 4.1 μm full width half maximum (FWHM) beam on the gap between

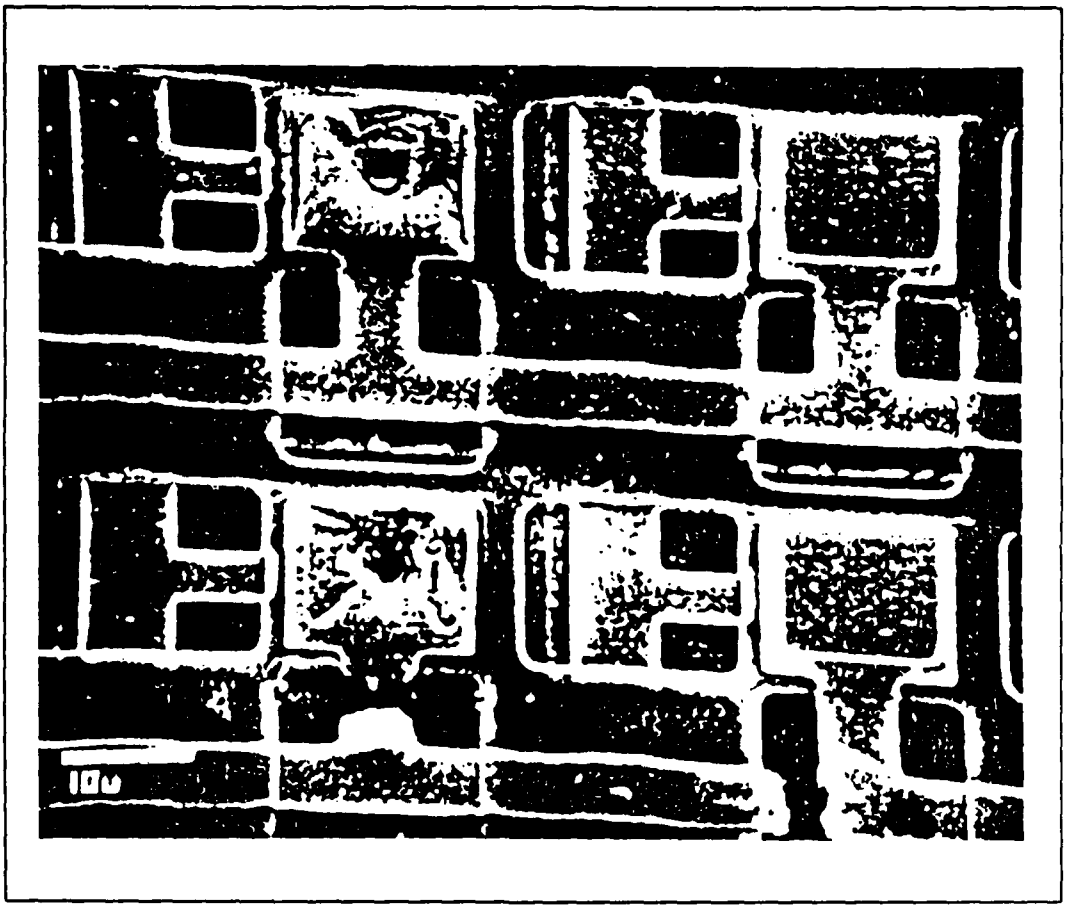


Figure 2.1. Lincoln Laboratories Fusable Vias and Cuts. [Lin85]

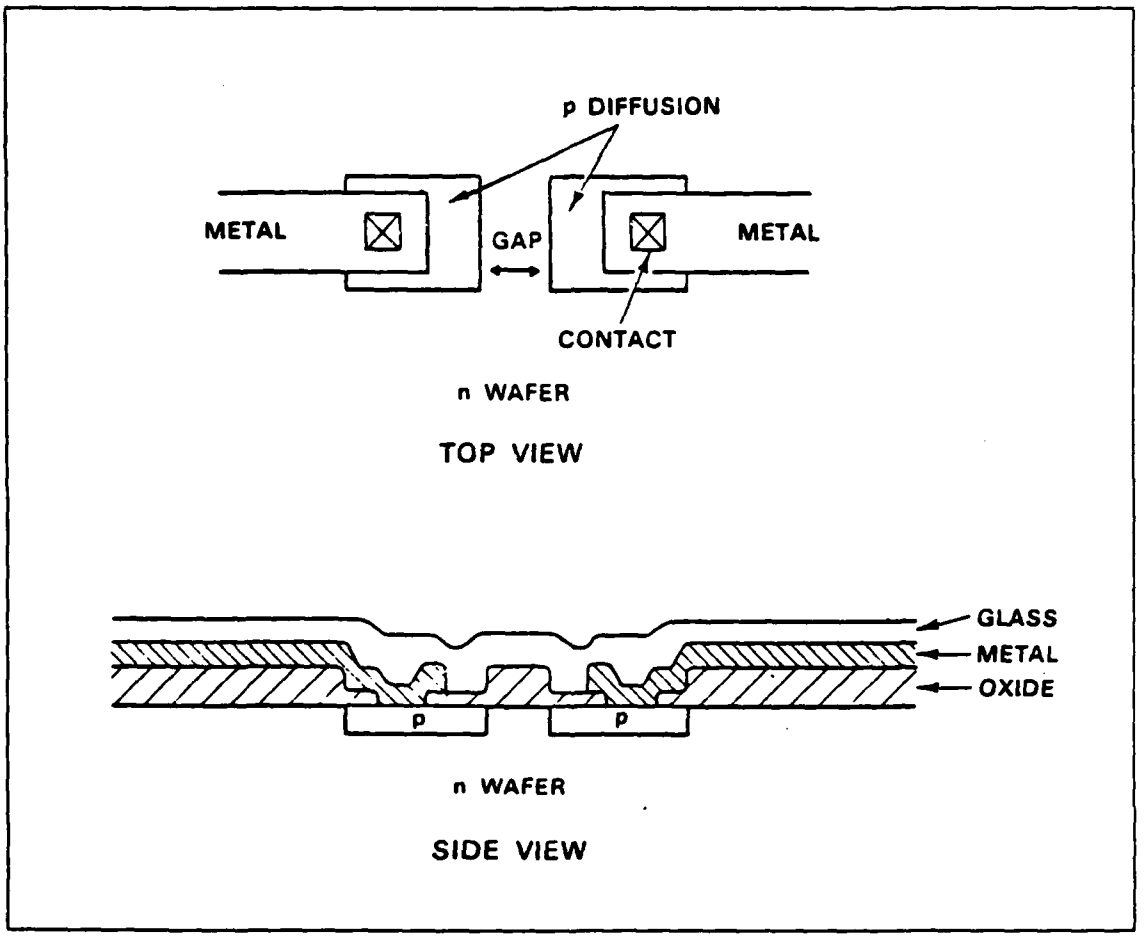


Figure 2.2. Lincoln Laboratories Fusable Links. [Lin88]

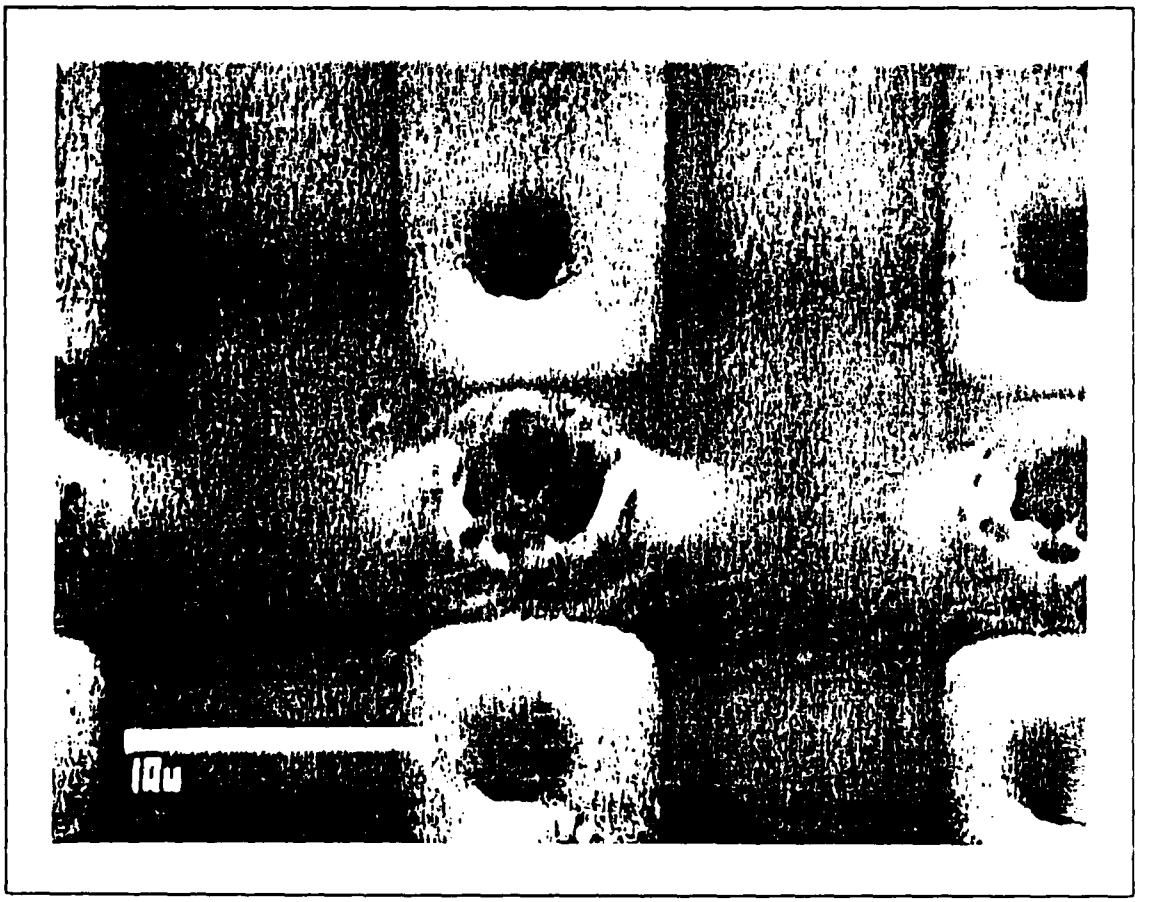


Figure 2.3. Photograph of Fusable Links [Lin88].

the diffusions. The localized heating at the gap causes the dopants in the two diffusions to diffuse into the gap thus forming an ohmic contact.

For λ=1.25 μm technology, the capacitive load of these unlinked diodes was measured to be 24x10^{-15} farads for n-type diffusions and 16x10^{-15} farads for the p-type diffusions. In either case, the pre-link resistance was measured as greater than one megohm. The linked resistance of the n-type links is approximately 60Ω and 150Ω for the p-type. These numbers are based on the dopant levels in the diffusion as a result of MOSIS processing. Acceptable positioning tolerances of the laser spot are $\pm$3 μm transverse to the gap and $\pm$1 μm along the gap [Lin86].

While this process may not offer the very low programmed resistance of the fusable via, it is 100% MOSIS compatible thus making it available to all MOSIS customers without the need for a separate contract with Lincoln Labs. SPICE analysis will determine if the fusable link resistance has a significant effect on operating speed if it is integrated into the AFIT LPROM.

2.2.2 **Laser Scanned CMOS LPROM Array.** Another form of RVLSI circuit uses a laser to change the physical characteristics of a diode to increase the reverse bias current. This current can then be used to bias transistors in a ROM on or off.

2.2.2.1 **Diode Biasing by Laser Scanning.** The initial research into this method occurred as a result of investigations into the effects of laser annealing on the silicon/silicon dioxide interface. Studies by V.G.I. Deshmukh, H.C. Webber and D.V. McCaughan using a ruby laser with energy densities to 1.1 J/cm^2 and pulse durations of approximately 13 ns showed that above .8 J/cm^2, physical deformations, in the form of wrinkles, occurred in the SiO_2 [Des81]. The most accepted explanation for this phenomena is that the laser energy radiates through the transparent SiO_2 and is absorbed by the silicon. Having a melting point of 1415^oC, the silicon will melt before the silicon dioxide with a melting point of 1700^oC. As it cools, the silicon will epitaxially recrystallize yielding some silicon which has deformed to match the lattice structure of the overlying oxide.

Similar deformations were not found in silicon structures which were annealed prior to application of silicon dioxide [Des81].

In developing laser programmable diodes, the laser is pulsed on a *p-n* junction. The generated bulk deformations in the surface of the silicon create hole/electron recombination centers for the thermally generated holes and electrons. The increased number of recombination centers decreases the minority carrier lifetime which in turn increases the reverse bias leakage current. Under reverse biasing, minority carrier leakage current is dominant current [Sze85].

2.2.2.2 **Diode Based CMOS LPROM Array.** Two members of Texas A&M University employed these effects in fabricating a 256 bit LPROM. Figure 2.4 and Figure 2.5 illustrate two configurations of memory cells used in their design. In the two diode/one transistor cell, two diodes are placed end to end with the anode of the lower diode tied to ground and the cathode of the upper diode tied to a 5 volt V_{dd}. The node between them is tied to the gate of the transistor. Biasing is performed by scanning one of the diodes. Scanning the upper diode will turn the transistor on while scanning the lower diode will ensure the transistor will not pass any current.

In the second design, the lower diode is replaced with a transistor. This transistor's gate voltage is held constant through the biasing of a third transistor whose gate voltage is in turn held constant by a laser scanned diode. The increased complexity of this cell is designed to provide greater stability of the bias voltage on the transistor. The increase in size is minimal since the minimum size CMOS diode is larger than the minimum size CMOS *n*-type transistor. In their design, minimum diode size was 10x20 μm, the minimum transistor gates were 5x5 μm. A supply voltage of 5 volts resulted in 130 ns read times and 160 ns precharge times [Lee87]. With either memory cell, the remainder of the addressing and output circuits are similar to the basic ROM. Greater density and quicker access times are possible with the newer MOSIS submicron fabrication technologies as compared to the 5 μm technology used in this effort.

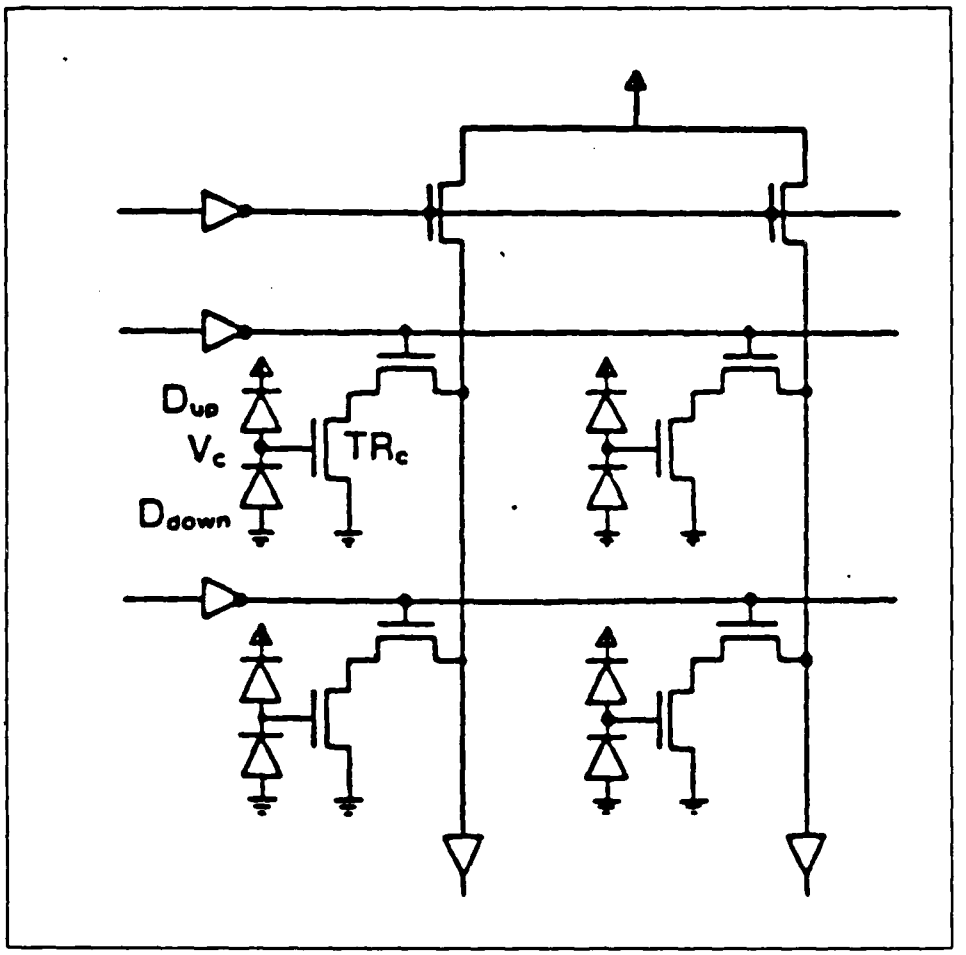


Figure 2.4. Two Diode One Transistor LPROM Memory Cell [Lee87].

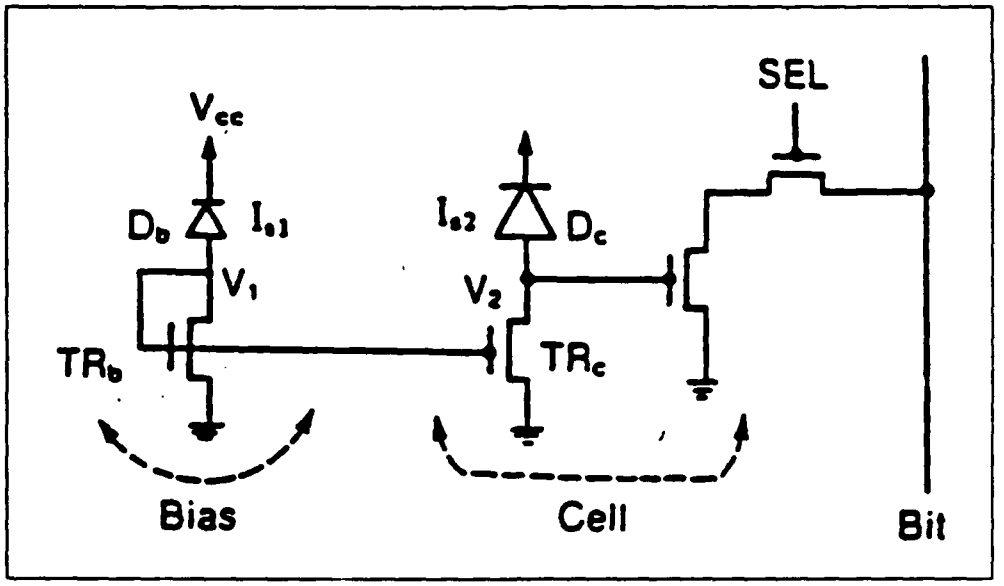


Figure 2.5. One Diode Three Transistor LPROM Memory Cell [Lee87].

Programming was accomplished using a Neodymium: Yttrium Aluminum Garnet (Nd:YAG) laser with a 6 μm spot size, less than 8 mW power and a 65 ns pulse duration. Figure 2.6 illustrates a completed 256 bit LPROM design [Lee87].

This LPROM design offers the advantage of requiring only a low power laser; however, even with a smaller fabrication technology, its operating speed is too slow and density too low to be a practical solution for the user programmable memory in FPASP.

2.2.3 **AFIT Laser Programmable Circuits.** In 1987, as part of his thesis effort, Capt. Craig Spanburg engineered a pair of laser programmable circuits and a programming station. His results and conclusions provide an excellent starting point for the development of the AFIT LPROM and its programming station. His research is presented here in three parts; the programming station, the automation software and the VLSI circuits themselves.

2.2.3.1 **Circuit Programming.** The programming station designed by Capt. Spanburg consisted of a Micro-Manipulator probe station with MITAS stepper motors and controller, a pair of lasers and a combination of lenses and mirrors to direct and focus the

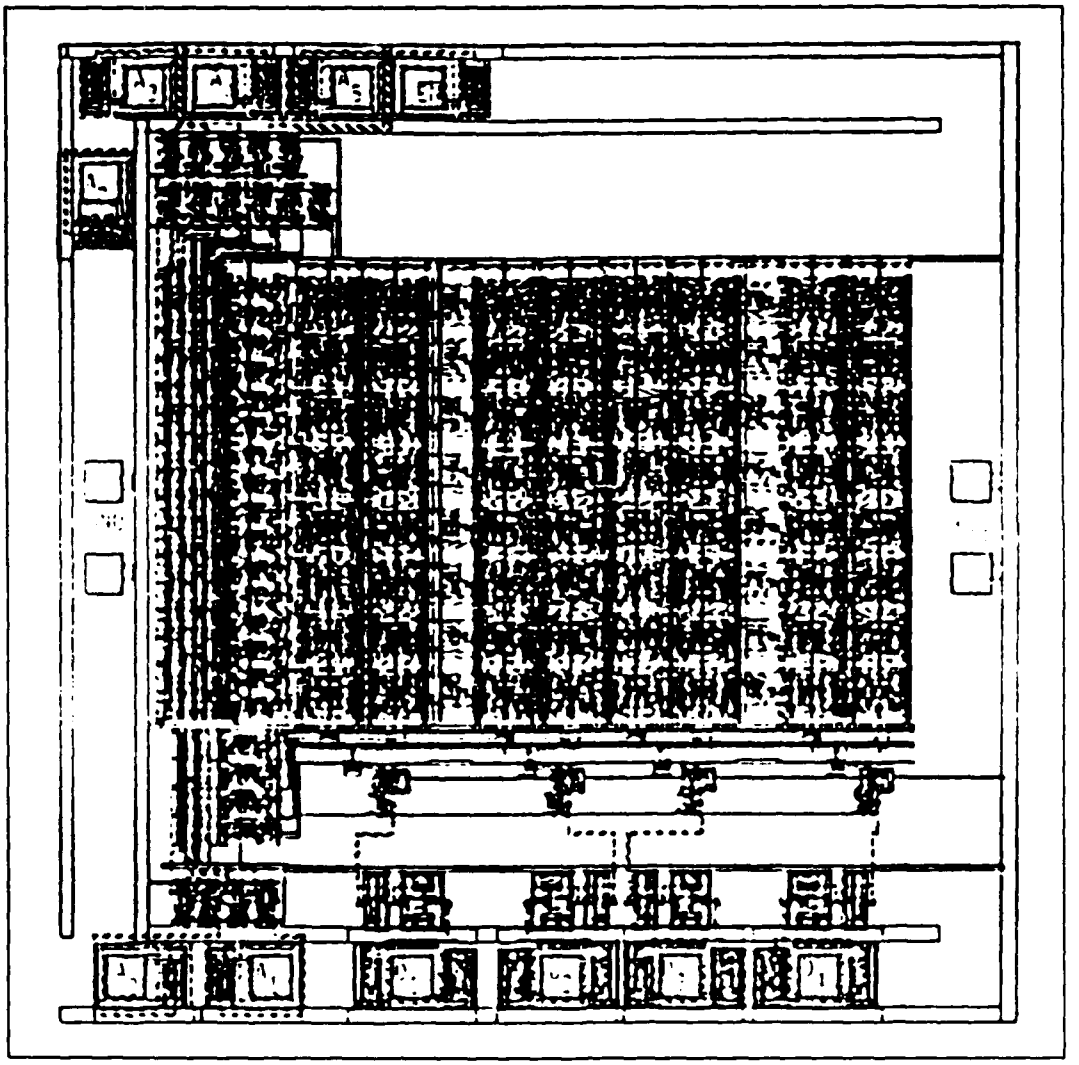


Figure 2.6. 256 Bit CMOS LPROM Array. [Lee87]

beams through the Micro-Manipulator optics port. A 50mW helium-neon laser is used to visually spot the desired blast point. Upon alignment, Nd:YAG laser is used to program the circuit. A beam splitter allows both lasers to be directed on exactly the same spot. The combination of the Micro-Manipulator stage and the MITAS stepper motors with their controller are capable of .714 μm steps with a 3% degree of accuracy [Spa87].

Hardware problems encountered by Capt. Spanburg include an inability to vary the power of the Nd:YAG laser. The power output of the laser, after it was focused down to the desired spot size, was above the energy necessary to program the circuits A rectangular aperture would have resolved this problem; however, none was available at the time of his effort. In addition, no beam expanders or collimaters were available and the available mirrors were uncoated for the frequency of the Nd:YAG laser. This resulted in the disfiguring of several of the aluminized, front surface, metal mirrors and the exposure of the circuits to an unconditioned laser beam. In addition, the belt drive system coupling the stepper motors with the X-Y translation stage was a source of positioning error. The slippage and stretching introduced by these belts was far greater than the 3% error introduced by the stepper motors themselves [Spa87, Sup87].

These results led to the purchase of a new Florod corporation laser optics box. It is a self contained unit equipped with many of the features recommended by Capt. Spanburg including a beam expander and collimater, microadjustable rectangular aperture and, dielectric mirrors capable of handling the an argon-ion or Nd:YAG laser. Direct coupling of the motors to the stage will eliminate the belt slippage.

2.2.3.2 **Circuit Programming Software.** The circuit programming software consists of manual control of the stepper motors through the keyboard and the ability to feed precalculated coordinates. The computer prompts the user for the coordinates row by row. The input is in the form of a decimal number which is then transformed into its binary equivalent. The relative placement of the *1's* and *0's* in the number determines whether a program point receives a laser blast or not. The user must input the length of each number in bits, the total number of words and the step distance between each program point [Spa87].

Due to the expected large number of program points in the LPROM, a more automated system needs to be developed to input the desired address and control the movement of the X-Y stage. An advanced feature would include the use of pattern recognition to improve placement accuracy and compensate for fabrication errors.

2.2.3.3 **VLSI Circuit Designs.** All of the circuits designed in Capt. Spanburg's thesis effort use the removal of metallization lines as the programming vehicle. In general, two metal lines are fabricated connecting a control node with both power and ground. Programming is accomplish by cutting the undesired connection. Figure 2.7and Figure 2.8 are diagrams of Capt. Spanburg's laser programmable 32-bit comparator and serial multiplier, respectively.

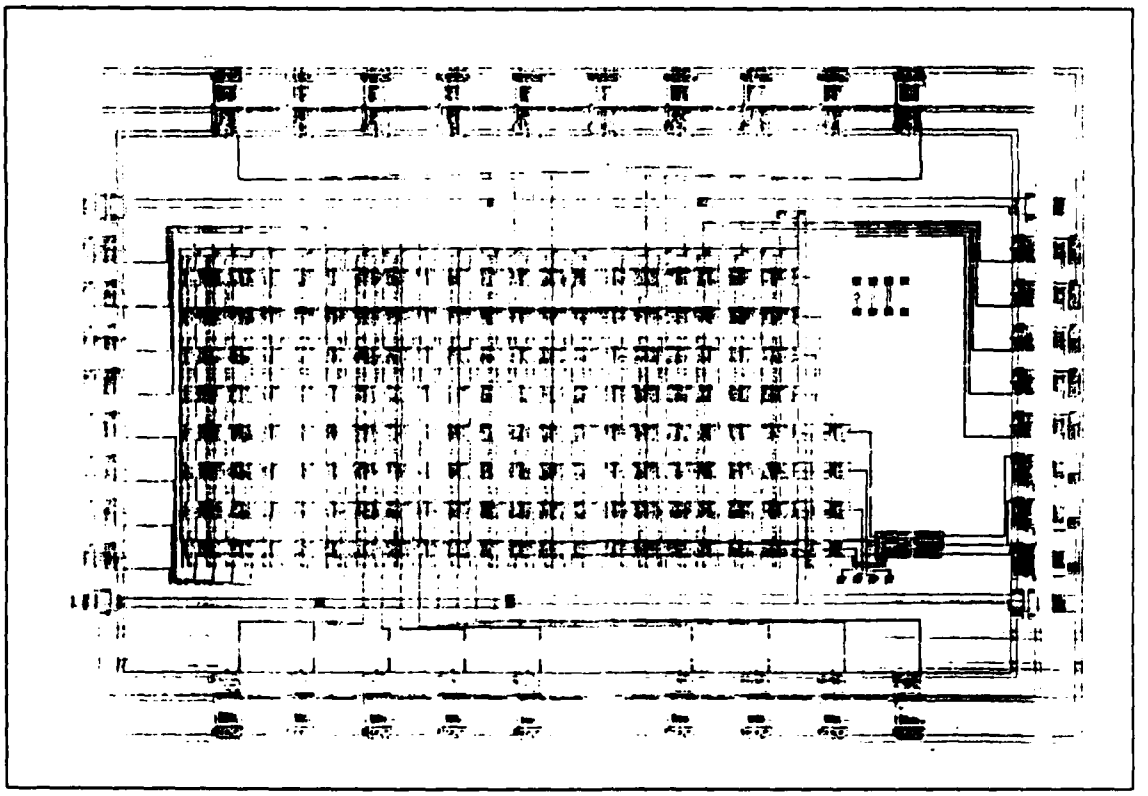


Figure 2.7. Laser Programmable 32-Bit Comparator [Spa87].

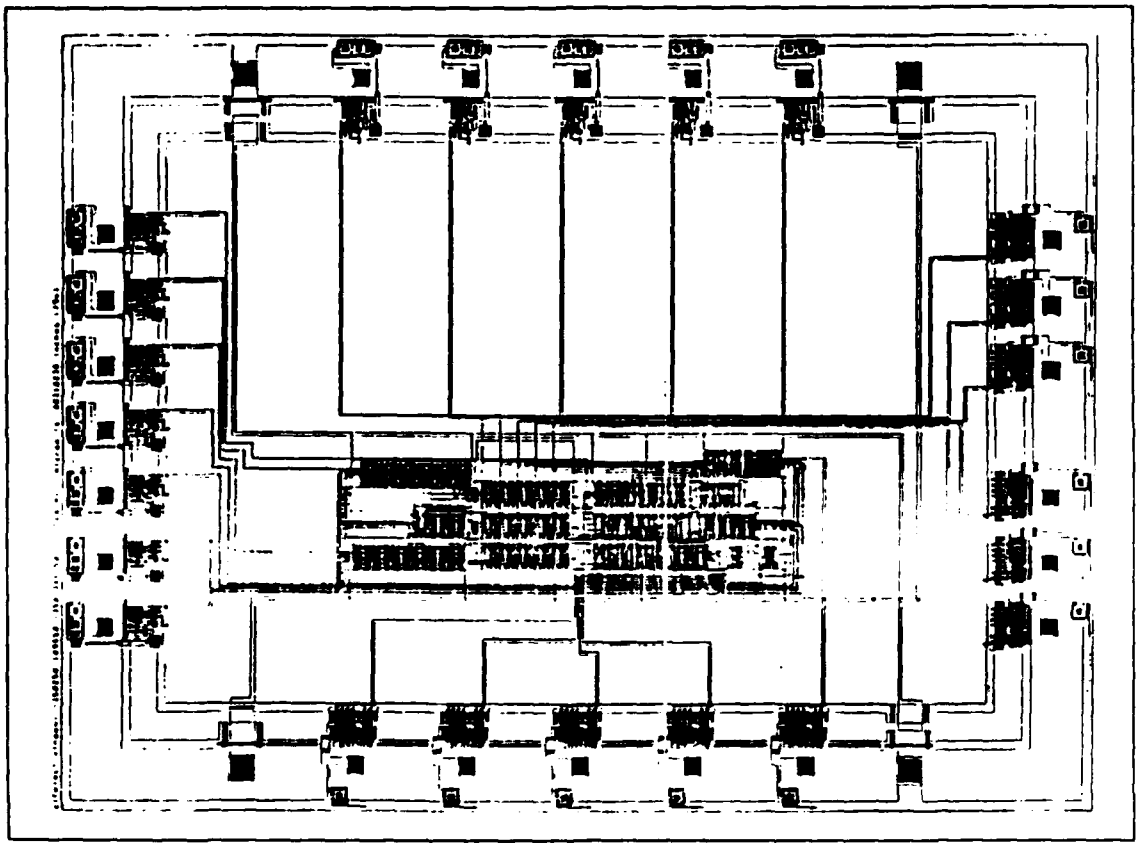


Figure 2.8. Laser Programmable Serial Multiplier [Spa87].

Due to the errors in placing the circuit beneath the laser, long, narrow, cuttable metal lines had to be designed into the circuits to ensure successful cuts. This forces an increase in the circuit's overall size. Because of these hardware problems, Capt. Spanburg was unable to successfully program these circuits using the equipment described above. The VLSI circuits were personalized using a Florod LFA laser cutting system on loan to AFIT for demonstration purposes.

2.3 VLSI Pattern Recognition Research

In 1988, Cpt. Erik Fretheim devised a means of reverse engineering VLSI circuits through a combination of pattern recognition and expert systems. The system uses a hardware and software configuration similar to that of Capt. Spanburg's. In his system, a Micro-Manipulator, with belt driven stepper motors, maneuvers the circuit beneath a multiple objective microscope and a video camera. A video image of a portion of the circuit is then captured, processed, and stored as a pixel array by an ITEX-100 video board [Ite87]. A cepstrum and fourier transforms are then taken of the image to determine its fabrication grid size or "λ", relative image strength and spectral density [Fre88]. The different signal strengths, which appear visually as different colors or shades of colors, signify different fabrication layers. Pattern recognition is then used to find vias and contacts between these layers after which a conductivity netlist is generated. An expert system then compares this netlist to determine the functionality of the area in view. The stepper motors step the circuit to the next adjacent view and the above procedure is repeated until the entire circuit has been scanned. It is then up to the expert systems to integrate the patchwork of functionality squares into an accurate description of the entire circuit.

The LPROM programming station will require many of the same features as this system. The same stepper motors and control software can be used to step from program point to program point. To ensure correct placement of the circuit beneath the objective, instead of searching for the circular shape of vias and contacts, the computer can be preprogrammed with the image of the program point which can then be compared, through pattern recognition, with the actual image of the circuit fed to the computer through the

the image processing board. Since the functionality of the circuit will obviously be known prior to computer scanning, the expert system will not be necessary.

2.4 XROM Design

L. A. Glasser and D. W. Dobberpuhl, in their book,"*The Design and Analysis of VLSI Circuits*", provide a detailed description of the precharging and sense amplification VLSI techniques essential to the quick operation of an XROM. These same precharging and sense amplification techniques are essential to maintaining acceptable operating speeds for the LPROM. Captain Paul Rossbach's article in the Custom Integrated Circuit Conference Proceedings, "An Optimizing XROM Silicon Compiler", provide a detailed description of the operation of the AFIT XROM [Ros85]. This information is valuable since many of the AFIT XROM design cells will be incorporated into the AFIT LPROM. Figures 2.9, 2.10, and 2.11 are illustrations of an AFIT XROM block diagram, an XROM memory cell, and an XROM sense amplifier respectively. The XROM block diagram shows the relative positioning of the sense amplifiers and multiplexers to the XROM cell array, the PLA decoders and the AO column drivers. The XROM cell diagram illustrates the relative placement of the word lines, *bitlines* and AO/AOBAR lines within a memory cell. As we will see, this particular diagram represents four memory locations all set to "1". The sense amplifier in Figure 2.11 illustrates connectivity only. To achieve the sensing effect, the switching voltage from "high" to "low" of an inverter is raised by increasing the ratio of the size of the *n*-transistor to the *p*-transistor.

2.4.1 **Address Decoders and AO Drivers.** The three least significant bits (LSBs) of the address bus, A0-A2, are stripped away from the rest of the address bus. The LSB A0 and its inverse A0BAR are used to select between the right and left column pairs of transistors. Consequently, these two signal lines are repeatedly propagated vertically for every two vertical columns of transistors. Bits A1 and A2 are fed into a 4 TO 1 multiplexer beneath the sense amplifiers to select which one of four *bitlines* is to sensed by the sense amplifier. This multiplexing of *bitlines* reduces the total number of sense amplifiers. These amplifiers are necessarily wide and can limit the density of the ROM array

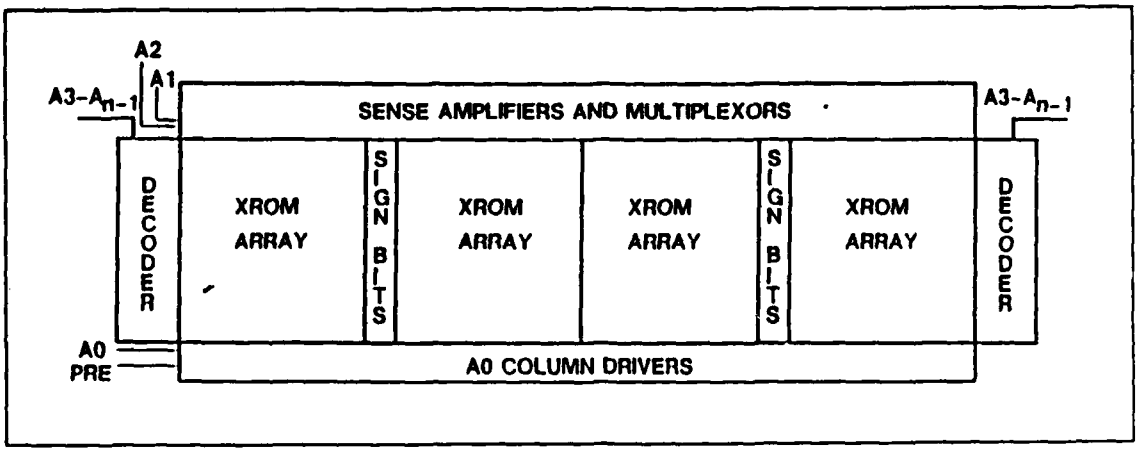


Figure 2.9. XROM Block Diagram [Gal87].

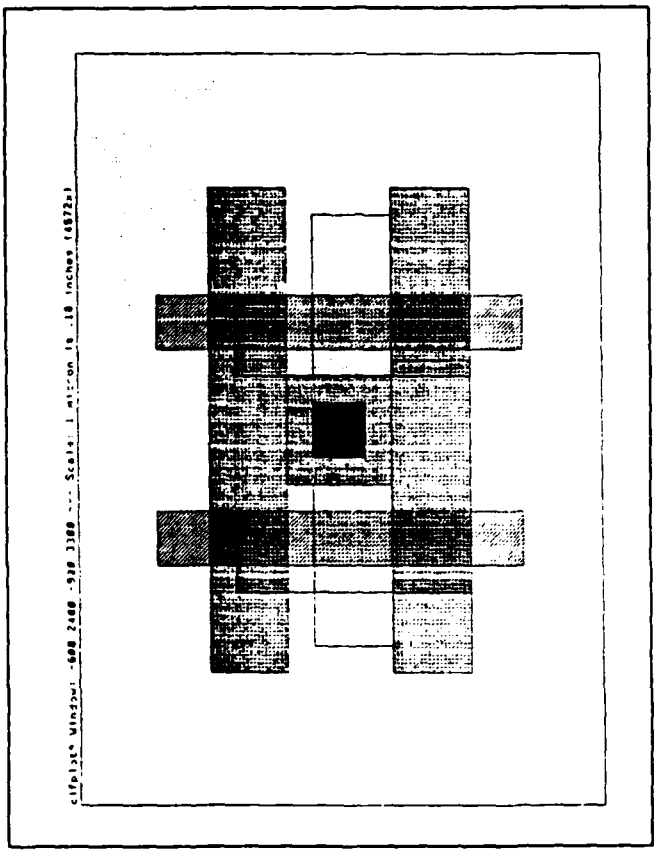


Figure 2.10. XROM Cell [Gal87].

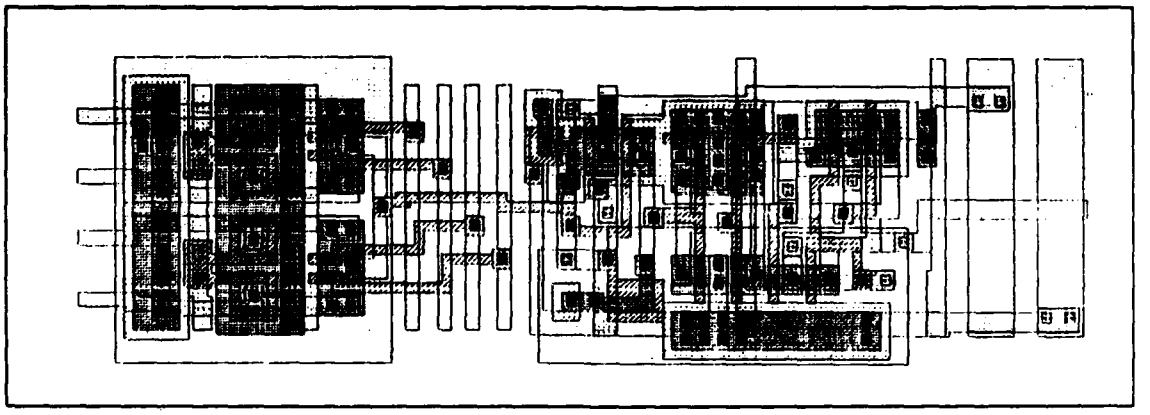


Figure 2.11. XROM Sense Amplifier and Multiplexer [Gal87].

if their inputs are not multiplexed. The remaining bits of the address are injected into a PLA decoder where they are decoded into individual, horizontally propagated, polysilicon *wordlines*. Due to the decreased overall density of the LPROM memory cell arrays, the A0/A0BAR lines are not practical and will be omitted. Similarly, due to the increased size of the LPROM cells, only two *bitlines* need be propagated per sense amplifier and the multiplexer reduced from a 4 TO 1 down to a 2 TO 1. Thus only one address line is wired to the sense amplifiers.

2.4.2 **XROM Memory Cell Array.** Every intersection of a *wordline* with either A0 or A0BAR and a *bitline* is a single bit memory location. The *metal2 wordlines* are contacted down to *metal1* which is in turn contacted down to polysilicon. This polysilicon forms the gates to transistors fabricated between the *bitlines* and the A0/A0BAR lines. To generate four memory locations all programmed as a "1", as seen in Figure 2.10, the drains of four transistors, forming an "X" shape, are all tied together at the *bitline*. The two left-most transistors' sources are tied to the A0 line while the two right-most are tied to the A0BAR line. The upper two transistor's gates are tied to the upper *wordline* and the lower two transistors' gates are tied to the lower *wordline*. The existence or omission of a transistor at these intersections determines if the memory location is to treated as a "1" or a "0".

The actual "1" or "0" signal to be sensed and output is on the vertically propagated *bitlines*. One *bitline* is needed for every one instance of an A0/A0BAR pair. These vertically propagated lines are usually fabricated in *metal1* to reduce resistance and increase speed [Gal87]. The LPROM's ones and zeros will likewise be sensed on the *bitlines*; however, the drains of the LPROM cells will be laser linked to a dedicated ground line rather than the A0 or A0BAR lines.

2.4.3 **Sense Amplifier and Bitline Multiplexer.** The XROM sense amplifier is designed to detect or sense any changes in this precharged voltage on the *bitlines*. The *p* and *n*-transistors, T_{sp} and T_{sn}, in Figure 2.11 are the primary sensing elements of the sense amplifier circuit. They function as an inverter whose input switching voltage is determined by the ratio of their respective drain to source current gains. The equations for determining the inverter switching voltage can be found in the aforementioned *Glasser and Dobberpuhl* text [Gla85].

Analysis of the XROM sense amplifier dimensions results in a theoretical switching voltage of approximately 3.8 volts. The sensing requirements of the LPROM will remain identical to those of the XROM, both employ precharged *bitlines* that are pulled down toward zero volts. SPICE analysis of the sense amplifier and multiplexer will determine if these designs are adequate. If not, only minor modifications should be necessary.

2.4.4 **XROM Precharging.** Programming of an XROM is performed by selective diffusion over the polysilicon *wordlines* to form the transistors that connect the A0 and A0BAR lines to the *bitlines*. To help compensate for the high resistance of the polysilicon lines, a low resistance *metal2* line is fabricated directly over the polysilicon line and periodic vias or interconnects are placed between these two conductors.

Most high speed ROMs, including the XROM, make extensive use of the *precharging* technique. *Precharging* uses an initial percentage of a clock cycle to permit specific electrical nodes to be charged to the same voltage as the supply voltage. The charge is passed through transistors that are clocked by a special precharge pulse. At the end of the precharge pulse, these transistors close the path from the supply voltage to the node. The charge is

then stored on the various internal capacitances of the node. In the case of the XROM, these might include the sources, gates, and drains of the MOS transistors, the capacitance between the metal or polysilicon lines and the grounded substrate. In other circuits, it may be stored on MOS capacitors built specifically for this purpose [Gla85].

2.5 XROM Operation

The sequence of events in reading a single bit from an XROM are discussed below. The reading of a whole "word" requires the same procedures be accomplished simultaneously, and in parallel for each bit in its respective column.

At the beginning of the clock cycle both A0 and A0BAR, and part of the sense amplifier are precharged to approximately 5 volts. The *bitlines* are precharged to 4 volts by a doughnut shaped, *n*-pullup transistor Simultaneously, the remaining bits of the address are being decoded by the NAND gates of a PLA decoder.

Next, after the 5 volt sense amplifier precharge signal is dropped, either A0 or A0BAR will remain set to 5 volts and the other will be dropped, through an inverter, to 0 volts depending of the given value of A0. It is important to realize, the selected line is the one that is set to 0 volts. The PLA will set the decoded *wordline* to 5 volts thus activating the gates of the transistors along its path. All other *wordlines* will be pulled down and held at 0 volts. If a transistor exists between the *bitline* and the A0 or A0BAR line set at 0 volts and this transistor is made active by charging its *wordline* to 5 volts, then the precharge on the *bitline* will be discharged through this transistor down toward 0 volts. If transistors exist between both A0 and A0BAR and the *bitline*, because they both share the same drain, these two transistors will fight with the resultant voltage being between 1.5 and 2 volts. The sense amplifier will detect this drop in voltage from 5 volts by means of an inverter with a high switching voltage, invert its output from "0" to "1" and thus this memory location will be interpreted as a "1". If there is no transistor, then the *bitline* will remain charged at 5 volts and the non-inverted sense amplifier output will be interpreted as a "0" [Gal87].

2.6 XROM Optimization

It is desirable to reduce the number of transistors in the XROM array to a minimum as each transistor represents both a load capacitance to the *word* and *bitlines* as well as a power sink of the precharge signal. Optimization algorithms such as those developed in 1985 by Captain Rossbach and Captain Richard W. Linderman, analyze each column to determine if it contains more *1's* or *0's*. For those columns that contain a large number of transistors, more than half the available memory locations, the *1's* and *0's* are inverted, the *1's* transistors removed and the *0's* inserted and the 2 TO 1 *wordsign* multiplexer selects the non-inverted sense of the column. This is the function of the two transmission gates at the top of the sense amplifier in Figure 2.11 [Ros85].

2.7 Summary

The following chapters discuss in detail how the engineering concepts first presented in Capt. Spanburg's thesis effort are expanded, reengineered and combined with various aspects of Cpt. Fretheim's reverse engineering station to create an automated laser programming station. The design techniques and actual layout of the AFIT LPROM, including SPICE simulations, are presented in Chapter 4. The unique aspect of this thesis effort is engineering the automated programming station and combining it with a sound VLSI device to create a practical and reliable VLSI circuit and tool.

III. VLSI Design and Fabrication

3.1 Introduction

The design processes begins with the development of the LPROM. The electrical characteristics of the Lincoln Laboratories fusable link must some how be integrated into a PROM structure which has a horizontal pitch that matches the AFIT XROM. This chapter begins with a brief discussion of the tools and technology files used to layout and fabricate the circuit. The VLSI circuit components and architecture are then presented. This is followed by SPICE simulations of the proposed design.

3.2 Layout and Fabrication

The VLSI software layout tool MAGIC, with MOSIS version 6 design rules for scalable CMOS installed, was used to layout the LPROM circuit.

Two fabrication technologies were used in designing the LPROMs. One LPROM was designed using MOSIS 2 μm *n*-well Tiny Chip fabrication technology, the other, a Hewlett Packard 1.2 μm *n*-well process. The MOSIS Tiny Chip is a 40 pin package with a maximum size of 2250 x 2180 μm including the input/output pads. At the end of the design process, an LPROM containing a 16 wide by 32 high array of memory cells was inserted in this package. Figure 3.1 is the resultant CIF plot. This translates to 128 addresses of 16 bit words. The Hewlett Packard 1.2 μm *n*-well process test circuits are "piggy backed" on other, larger circuits such as the 16 and 17 Point Winograd Fourier Transform circuits. Each LPROM circuit had to be designed to fit in the available unused spaces.

Two potential problems worth noting relate to the inconsistencies in fabrication between MOSIS vendors and with the circuit itself. The relative positioning of features on a circuit as laid out in MAGIC may become distorted during fabrication due to bloating, undercutting or mask alignment errors. The density of the LPROM arrays are directly proportional to these errors which are documented in Chapter 5. These errors also have a direct impact of the design rules established for the generation of new LPROM circuits.

Also, Lincoln Laboratories has indicated that the diffusion wells of very small laser links may provide an insufficient amount of dopants or charge carriers to successfully

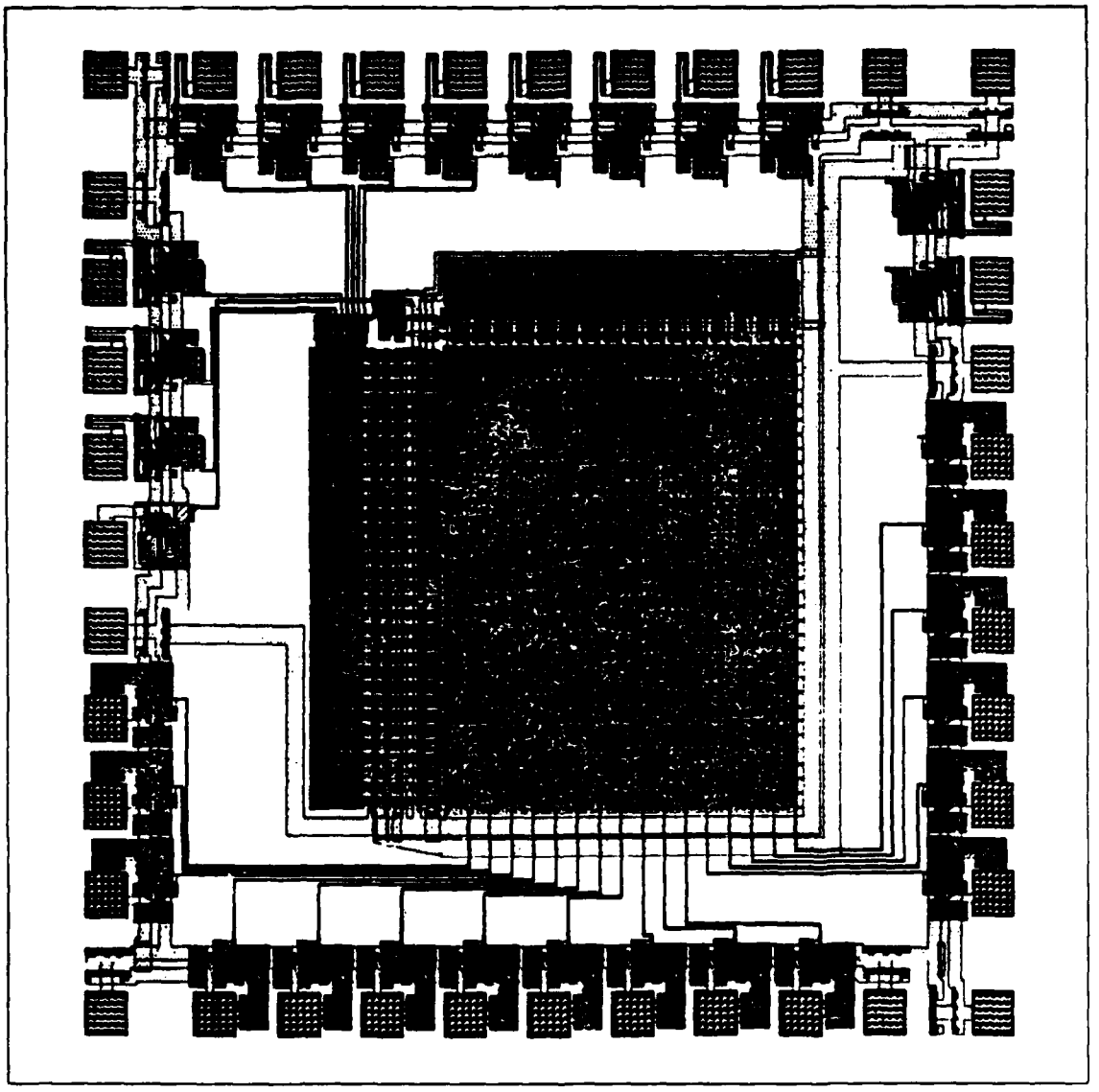


Figure 3.1. CIF plot of 2 μm 2K LPROM Tiny Chip.

electrically bridge the gap. This is especially true with submicron technologies where doping concentrations are light to begin with [Lin88].

3.3 LPROM Design

As mentioned in Chapter 1, the LPROM architecture will be based on the proven XROM. The major VLSI cells which comprise the AFIT XROM, detailed in Chapter 2, are the XROM memory cell array, the address decoders and A0/A0BAR drivers and the sense amplifiers and *bitline* multiplexers. Since most of the immediate applications of the LPROM, including the FPASP, involve coupling it to an XROM, the developed design must permit the LPROM and XROM to be addressed together as one whole memory unit. Ideally, to make use of all the existing XROM components, the LPROM cell dimensions should be identical to those of the XROM cells. However, this is not possible due to limitations in the accuracy of positioning the laser and the physical requirements of the laser link.

Like the XROM, the LPROM can be divided into three main components; the LPROM memory cell array, the address decoders, and the sense amplifiers and *bitline* multiplexers. The latter two components are essentially identical to those of the XROM.

Following a hierarchical design for this LPROM, the individual LPROM memory cells are first laid out in an appropriate sized array to form a single array cell. The sense amplifiers pairs are likewise arrayed and saved as a single sense amplifier array. A maximal size array of AND plane decoders containing 128 cells was laid out and sequentially addressed from 0 (00000000)to 255 (11111111). The user need only copy this array and select the appropriately addressed cells to suit his needs.

3.3.1 **LPROM Memory Cell Design.** Using scalable CMOS dimensions, the standard XROM cell described in Chapter 2 is 19 λ high by 26 λ wide and contains a total of four memory locations; two memory locations side by side for two vertically stacked rows of bits. A pair of these cells, placed side by side, is used for every column. This 52 λ column width is a function of the width of the memory cells. Four cells are partially overlapped for every column after which their effective width is 13 λ. The use of A0 and

A0BAR lines and a 4 TO 1 multiplexer to feed the sense amplifier creates an effective 8 TO 1 multiplexer. Each column can therefore sense one of eight bits for each row of bits.

With these limitations in mind, an LPROM cell was designed that was 52 λ wide 33 λ high which contains four memory locations divided into two horizontal bits for two vertical words. While being less dense vertically than the XROM, this design permitted all of the AND plane address decoders to be placed on one side of the array. This allows only one set of address lines to be run. The vertical density of the XROM arrays forced the AND plane address decoders to be placed alternately on either side of the array requiring two sets of address lines.

To function like an XROM cell, the LPROM cell has to provide a means of pulling the precharged *bitline* down to zero volts if the memory location is to be programmed as a "1" and allow the *bitline* to remain high at approximately 5 volts if the memory location is to be considered a "0". As in the XROM, the AND plane address decoders charge the *wordline* of the selected address to 5 volts. The *wordlines*, in turn, are shunted down to polysilicon which forms the gate of the memory location transistors. There is sufficient room in the LPROM cells to permit a separate *polyshunt* from *metal2* to polysilicon for each cell. This reduces the resistance of the *wordline* and decreases the rise time of the *wordline* to 5 volts. If the *bitline* side of the transistor is considered the drain of the transistor, the laser programmable laser link is formed on the source side by extending the diffusion away from the gate. A 3 λ gap is then formed followed by a renewal of the diffusion. This diffusion is then shunted, through *metal2* to ground. Thus, if the link is formed by a laser zap, the source will be tied to ground and the *bitline* will be pulled down toward zero volts if the transistors word line and gate are selected. Otherwise, the source will float and only a minimal charge will be passed through the gate; just enough to satisfy the parasitic capacitance of the floating drain. SPICE analysis revealed this is an insufficient amount to be sensed by the sense amplifier. Figure 3.2 is a CIF plot of this memory cell.

Due to the greater charge carrier mobility and reduced resistance, all diffusions are of *n*-type. All vertical lines are fabricated in *metal1*. These include the *bitlines* and the multiplexed output signals from the XROM and LPROM sense amplifiers. All horizontal

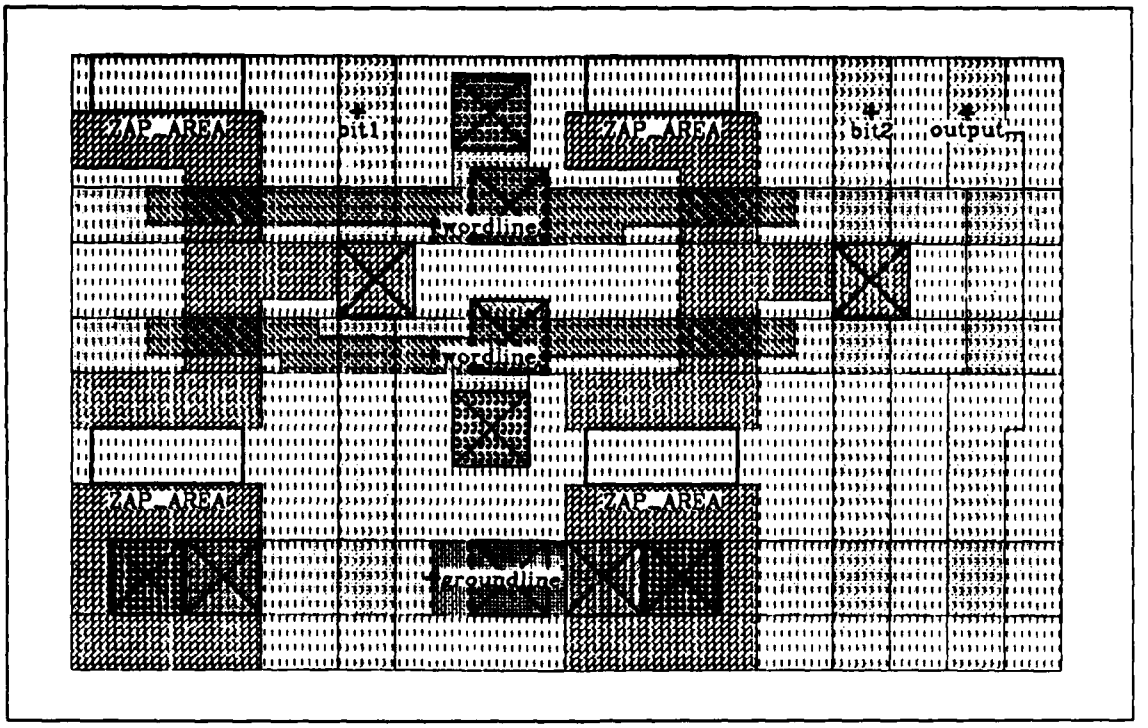


Figure 3.2. CIF plot of LPROM Memory Cell.

lines are run in *metal2*. These include the *wordlines* and the ground bus line. The *wordlines* are shunted down to polysilicon and the ground lines are shunted down to *n*-diffusion. One additional feature is a *p*-well contact made in each of the LPROM cells.

Given the restraint that the horizontal pitch of the LPROM must match that of the XROM, this memory cell represents the most dense design possible using version 6 MOSIS design rules [Mos88]. If the XROM's pitch is not a limiting factor, density could be increased by removing the *p*-well contact and the shunt from *metal2* down to polysilicon and narrowing the width of the *n*-diffusion. If the shunt to diffusion is removed the *wordline* signal will have to transmitted in polysilicon with shunts to *metal2* at the beginning and end of each array, much the same way it is performed in the AFIT XROM. All vertical dimensions are already minimized.

3.3.2 **Sense Amplifier Design.** As mentioned earlier, the sense amplifiers are based on those designed for the XROM. The differences are that the 4 TO 1 multiplexer at the bottom of the XROM has been reduced to a 2 TO 1 for the LPROM and the wordsign selector, used in the optimization routines of the XROM, has been replaced by a 2 TO 1 multiplexer to select between the outputs of the XROM and the LPROM. To facilitate diffusion well sharing between adjacent sense amplifiers, the sense amplifiers are laid out in mirrored pairs. In addition, a signal line is run in *metal2* back down through the sense amplifier and *bitline* multiplexer to the signal output lines of the LPROM memory cells. Because of the require regularity in the array of LPROM memory cells, the output signal line from the sense amplifiers is always located on the right side of the cell. This forced a redesign of the single sense amplifier into a left and right version to accommodate this signal line. The CIF plot in Figure 3.3 illustrates a sense amplifier pair.

3.3.3 **Addressing Decoders.** The AND plane address decoders used in the LPROM are identical to those found in the XROM with the exception of the shunts from *metal2* down to polysilicon having been removed. Each AND plane address decoder cell decodes two eight bit addresses and controls two *wordlines*. Figure 3.4 below is a CIF plot of an address decoder cell. Once again, since optimization of the memory cell array is not possible at the time of fabrication, a separate A0/A0BAR array is not necessary. For the same

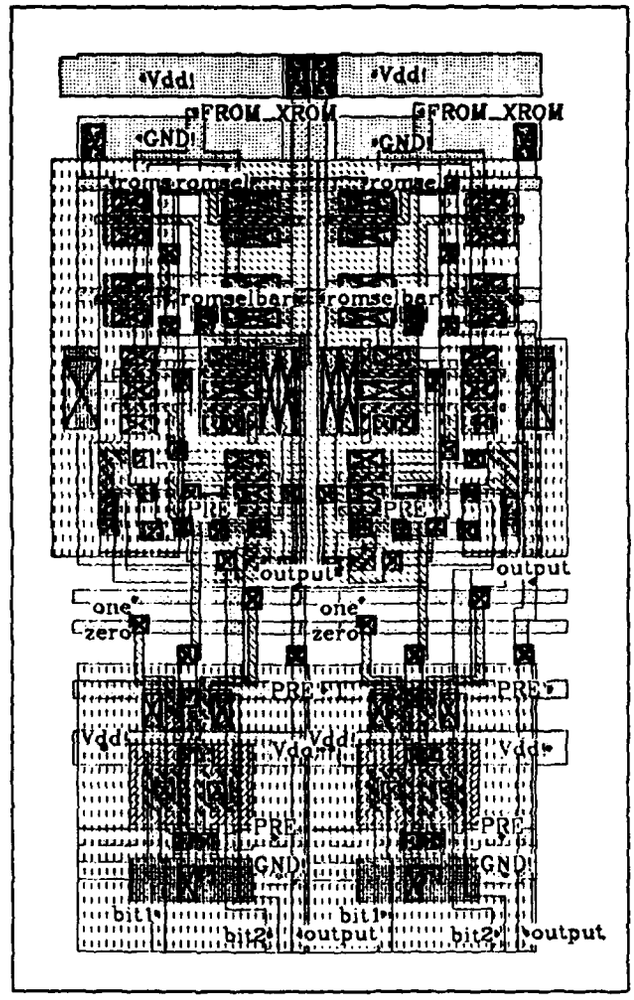


Figure 3.3. CIF plot of a LPROM Sense Amplifier Pair.

reason, the AND plane address decoders can be personalized at the time of fabrication with the addresses being sequential starting from all zeros. As an aid to the formatting of the data to be programmed and for the other laser programming algorithms, the lower left hand corner of the memory cell array is designated as the most significant bit of the word with the lowest, all zeros, address. In addition to the address lines, and their complements, which are injected into the AND plane address decoders, one additional address line is used to control the 2 TO 1 *bitline* multiplexer at the base of the sense amplifiers.

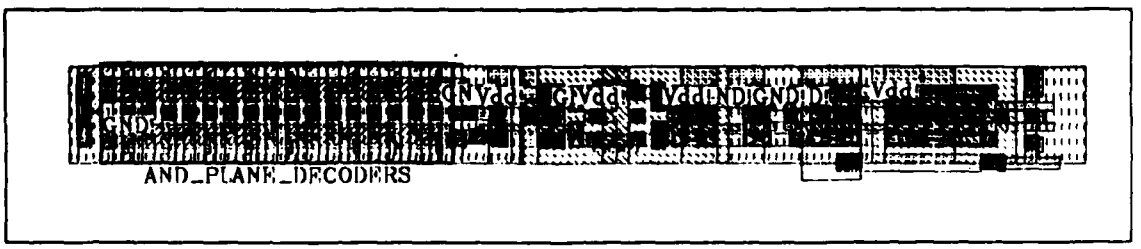


Figure 3.4. CIF plot of an And Plane Address Decoder.

3.4 SPICE Analysis

To ensure compatible operating speeds with the XROM, SPICE analysis was performed on the entire LPROM circuit. Using the *floating point application specific processor* (FPASP), clocked at 25 MIIz, as a metric, the LPROM circuit must be able to decode the incoming address, drive the appropriate *wordline*, pull down the *bitline*, have this *bitline* sensed and drive the output line to the memory register/drivers in time for the these registers to latch the output before the fall of the second phase of the clock cycle, Φ_2. In other words, the LPROM circuit has approximately 35 nsec to generate its output measured from the beginning of the clock cycle. The 10 nsec long positive sense of the first half of the clock cycle, Φ_1 is used as the precharge signal to the sense amplifier and the *bitlines*. The following diagrams form a schematic of the circuit used in the development of the SPICE deck. The circuit in Figure 3.5 simulates the AND plane address decoder and the *wordline*. The circuit simulating the programmed memory cell, *bitline*, *bitline* multiplexer

and sense amplifier is shown in Figure 3.6. The SPICE deck, which includes the actual sizes of the transistors and technology files, is included in Appendix A of this report.

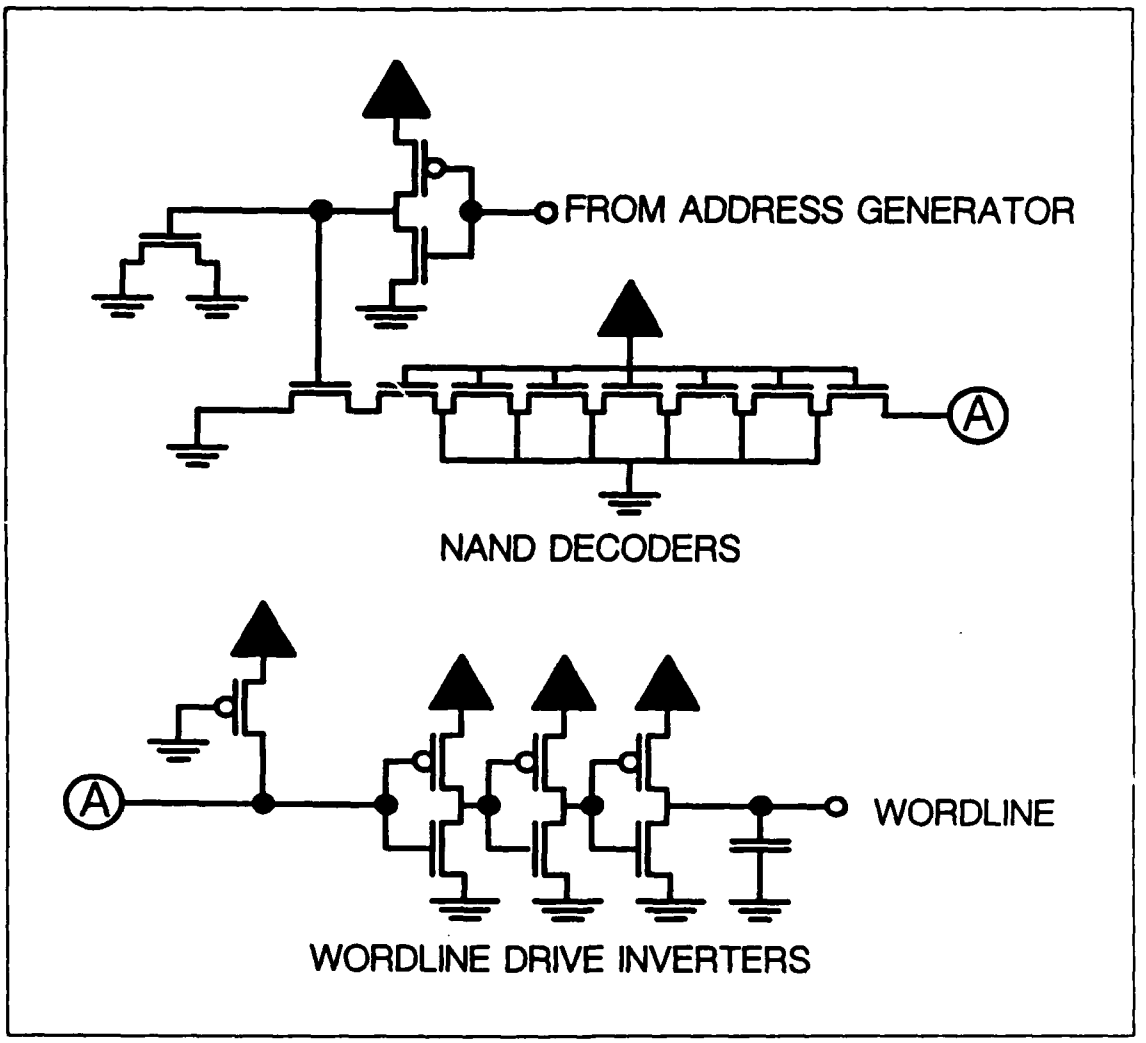


Figure 3.5. SPICE Schematic of the Address Decoder and Wordline.

The laser link is simulated by a resistor between the memory location transistor and ground. For the λ=1.2 μm simulation, a value of 1000Ω was used. A lower value of 100Ω was used for the λ=2.0 μm simulation. Both of these resistance values were experimentally determined and reported by Lincoln Laboratories [Lin88].

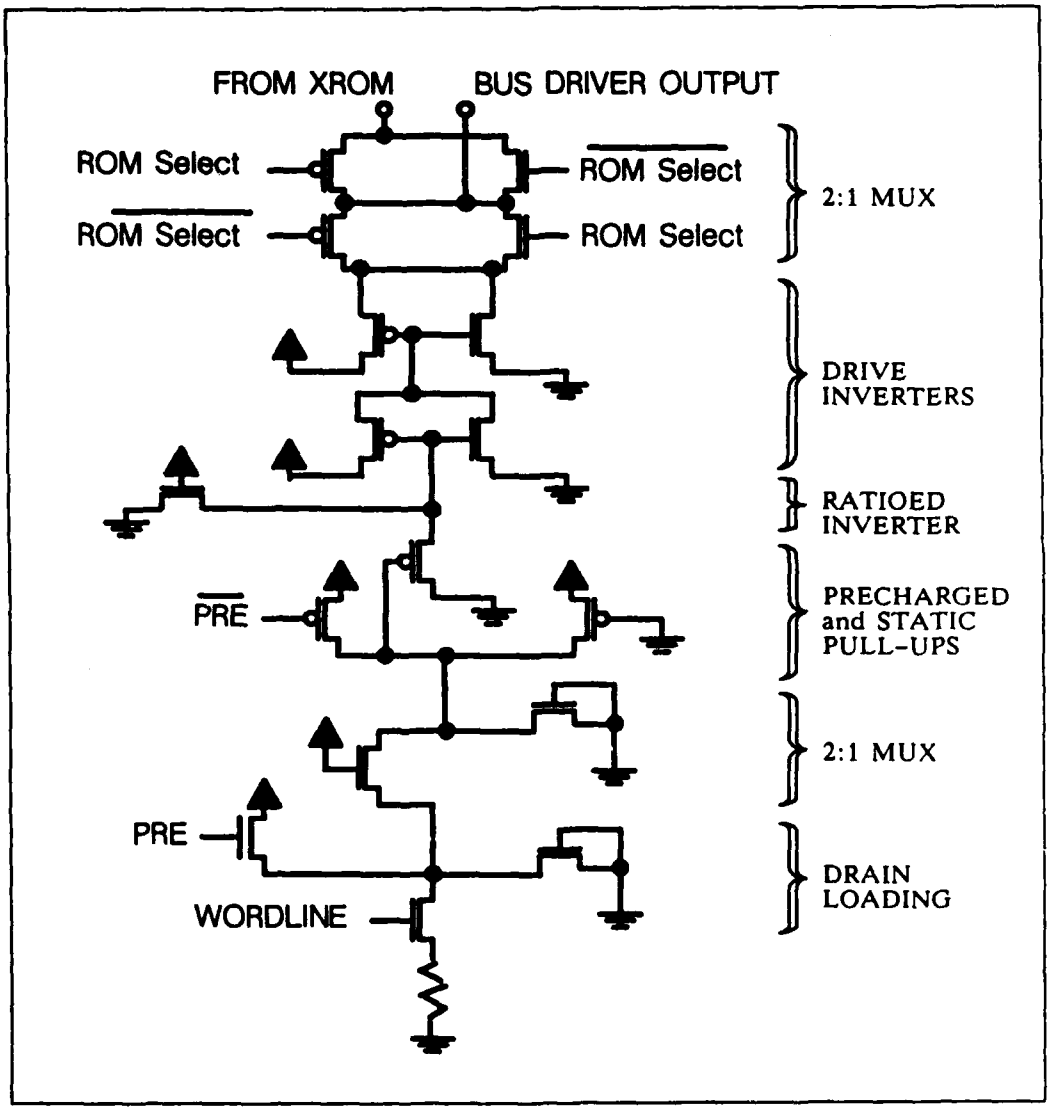


Figure 3.6. SPICE Schematic of Bitline, Bitline Multiplexer and Sense Amplifier.

The results for the λ=1.2 μm and λ=2.0 μm SPICE simulations show the output line crossing through 2.5 volts at 25 and 24 nanoseconds respectively. This analysis also reveals that a *wordline*, previously selected and charged, is driven back below the memory location transistor turn on voltage by the precharge signal prior to the end of this signal.

3.5 Summary

The critical component of the LPROM is the diffusion link, the remaining circuit designs have already been proven in the AFIT XROM. Within the guidelines established by the AFIT XROM and the version 6 design rules, this VLSI design represent the greatest circuit density possible. Consequently, the LPROM memory cell presented here assumes a high adherence to the CIF description forwarded to MOSIS. The SPICE simulations indicate that the speed of operation of the LPROM is comparable to the AFIT XROM and that the two circuits should function effectively together.

IV. Laser Programming Station Design

4.1 Introduction

The engineering of the LPROM programming station can be divided into hardware and software. This chapter first analyzes the proposed physical components of the programming station and their configuration. This is followed by a discussion of the software routines and how they must interact to control and automate this hardware.

4.2 Laser Programming Station Components

The design of the laser programming station is derived from a laser table built by Lincoln Labs. Semi-annual DARPA reports published by laboratory personnel as well as electronic mail and telephone conversations with Lincoln Labs were used to establish the basic design [Lin85, Lin86, Lin88].

4.2.1 **Four Watt Argon Ion Laser.** Laser linking and cutting can be accomplished with a 3.5 watt laser with at least a 70 μsec duration. This requires the use of a continuous wave laser. To date, pulse lasers are capable of no more than 10 μsec pulses. Lincoln Labs experimentally determined 513 nanometers to be the optimum wavelength. An argon-ion laser meets these requirements. Specifically, either the Spectra Physics 2020-04 or 2016-04 argon-ion laser, as shown in Figure 4.1, will be used. Laser power can be set either manually, or by computer through an RS-232 port. Lincoln Laboratory uses a Spectra Physics 2020 in their present configuration.

4.2.2 **Optics Box.** The Florod Corporation LFA laser optics box will deliver the laser beam to the circuit. Figure 4.2 indicates the useful features of this piece of equipment including; a microadjustable rectangular aperture which controls the laser spot size, a spot marker light which is directed through the aperture to aid in positioning the laser beam, a stand for the pattern recognition video camera, and high resolution objectives for viewing the circuit and final focusing of the laser beam. A beam collimater further conditions the laser light to ensure uniform power density throughout the exposed beam. Optical notch filters are placed between the camera and the laser light source. This will prevent

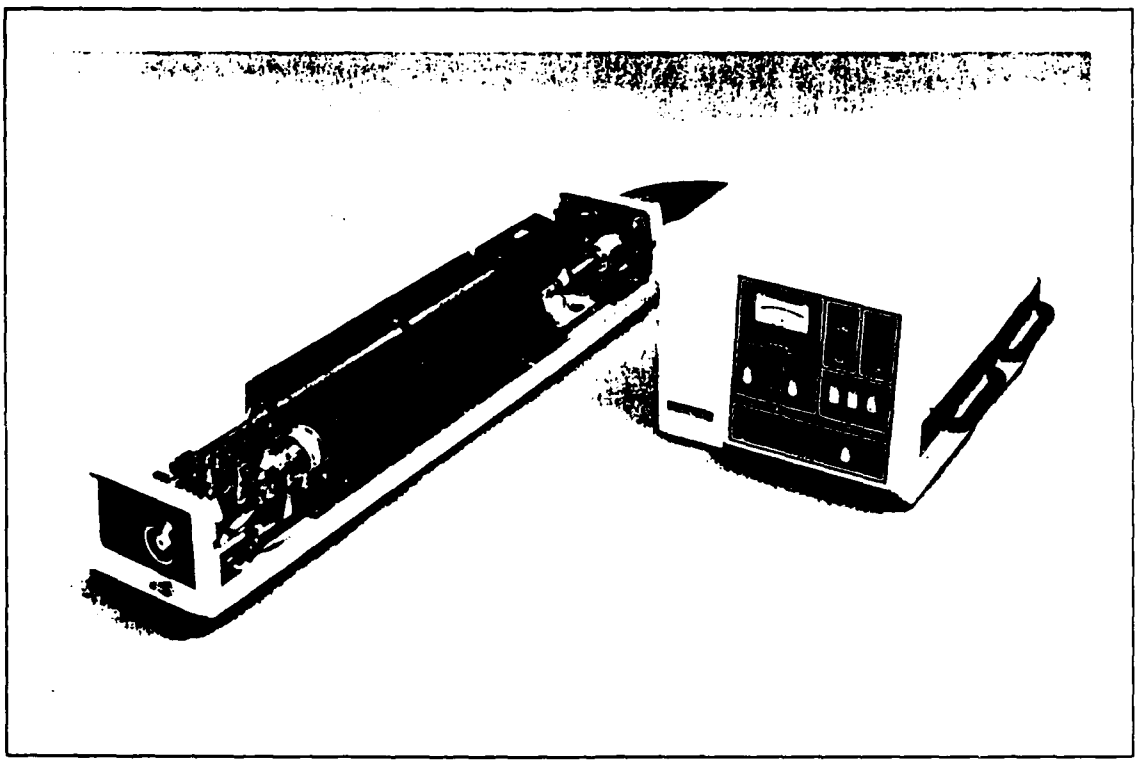


Figure 4.1. Spectra Physics 2020 laser.

the intense laser light reflected off the circuit from damaging the camera's charge coupled device and to increase the contrast between the different layers in the circuit [Fre88].

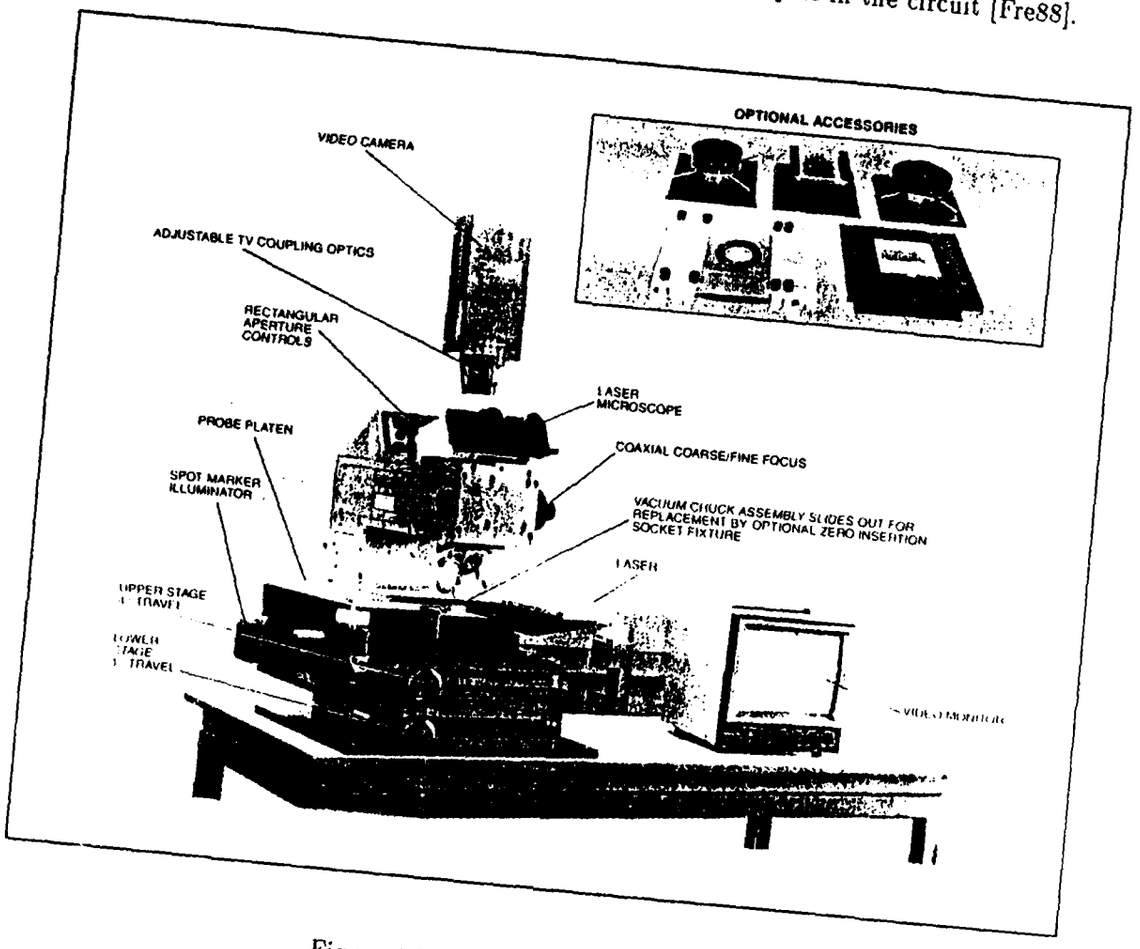


Figure 4.2. Florod LFA Optics Box.

4.2.3 **Shutter.** A Uni-Blitz guillotine shutter will form the first stage of a two stage shutter. According to the manufacturer, Vincent Associates, the fastest open and close time, measured from 50% open to 50% closed, is 1.5 msec. The surface of the 6mm shutter is coated with a highly reflective aluminum-silicon oxide which is capable of dissipating 6 watts per square millimeter of laser energy [Vin87]. This is important since the shutter will be exposed to the laser most of the time. By offsetting the incident angle of the laser to the shutter, the reflected laser energy can be directed into an energy absorption

pulse duration	*radius*	*cut length*
70 μsec	4.5"	0.1187"
80 μsec	4.25"	0.1282"
90 μsec	4.0"	0.1357"
100 μsec	3.75"	0.1414"

Table 4.1. Pulse Duration Timing Slots.

box. Since this type of shutter is incapable of reaching the approximately 70 μsec pulse length time, a second stage is necessary. The second stage consists of a rotating disk on a constant rotation motor. At a constant RPM, the arc length of the slit or slits cut into the disk will determine laser pulse duration into the optics box. A second timing slot made in the rim of the disk will trigger an LED/photodiode pair. This signal in turn feeds the synchronization circuit diagrammed below. The accompanying timing diagram illustrates how this circuit will output only one shutter trigger pulse for each computer generated pulse; even if the computer pulse overlaps two photo diode pulses.

The output of this synchronization circuit will provide the trigger signal to a model SD-10 Uni-Blitz shutter controller. A continuous duty motor of 3600 rpms will make one revolution in 16.67 μsec thus the first shutter needs to be open for some duration just less than this value. The table below lists the position and size of the pulse duration slits cut into this disk. Figure 4.4 illustrates the shutter apparatus.

4.2.4 **X-Y Translation Stage.** Initial designs call for a high precision Anarad X-Y translation table similar to the one in use at Lincoln Laboratories. This table is linearly encode in the X and Y directions and is equipped with laser interferometers which enable placement accuracies down to two tenths of a micron. Positioning data is supplied to the table through an RS-232 port. When it was determined that the price of this table was prohibitive, an alternate design was generated using the X-Y translation stage supplied with the optics box and computer controlled stepper motors. Pattern recognition software is integrated into this system to maintain the required positioning accuracy.

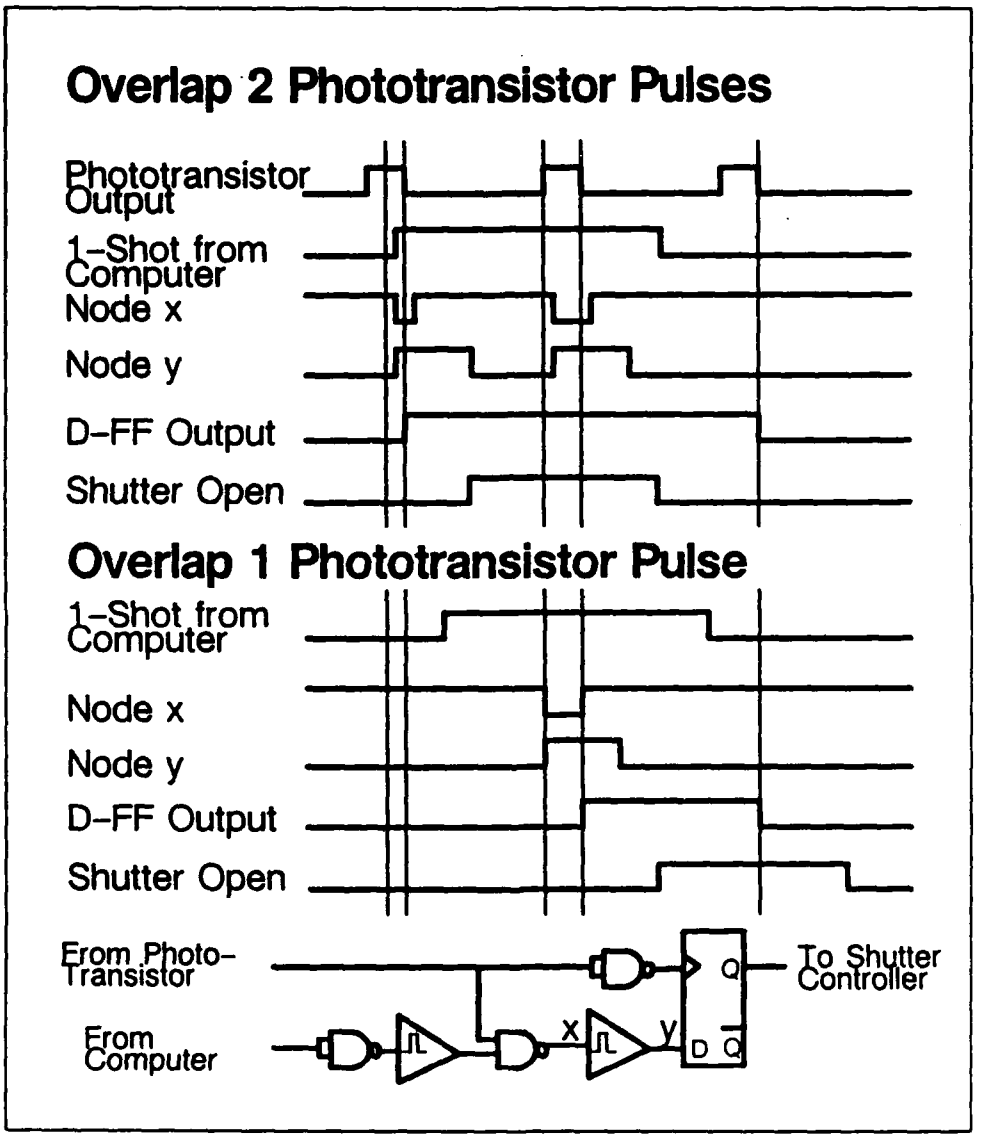


Figure 4.3. Shutter Timing Circuit and Diagram.

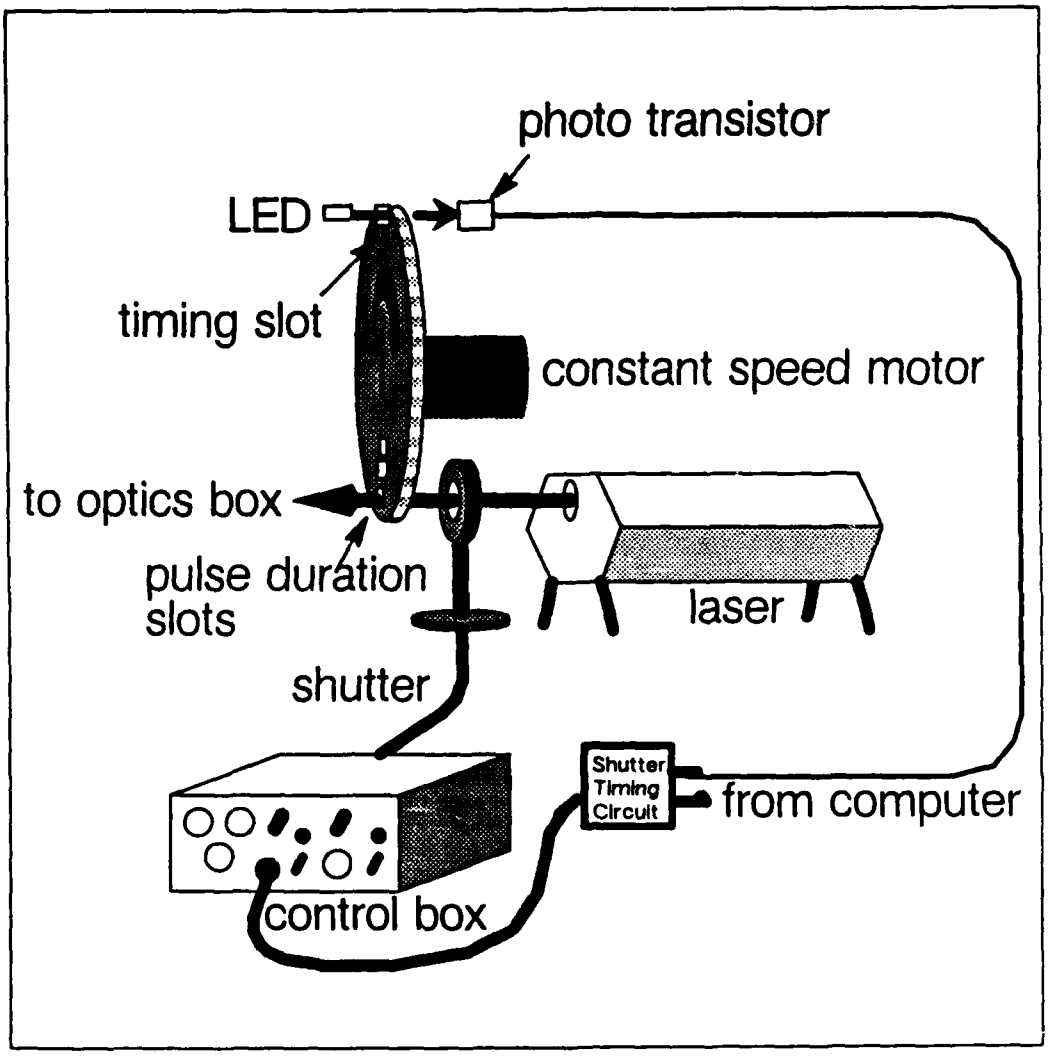


Figure 4.4. Shutter Configuration.

The Alessi MS-106 X-Y translation stage supplied with the Florod optics box uses manually operated adjustments with a maximum movement in both the X and Y directions of one inch. This is sufficient for work on a single circuit. A Superior Electric Modulynx controller will interface between the computer and the MITAS stepper motors directly coupled to the manual adjustments. This controller and motor combination is capable of 12,800 steps per 360° rotation. The final step size resolution will be experimentally determined after coupling the motors to the stage. The slippage and stretching in the present belt driven translation system has demonstrated the necessity for some form of direct coupling between the stepper motors and the translation stage's adjustment knobs. The Modulynx controller receives the X-Y translation instructions from the computer through an RS-232 port. Camera video will be ported into the host computer through an RS-110 port to an ITEX FG-100 image processing board which will provide input for pattern recognition software which, in turn, will provide final fine adjustment instructions to the MITAS controller.

The entire system will reside on an nitrogen suspension, anti-vibration table to reduce the number of mechanical errors induced by the environment.

4.2.5 **Computer Support.** The MITAS X-Y controller, the ITEX FG-100 image processing board, and the pattern recognition software are resident on a MicroVaxII computer system. A concurrent effort is being made to install this controller and software on a SUN4 workstation.

4.3 Automation Software Development.

The software development can itself be divided into three efforts. The first is to retrieve the data to be encoded and format it to meet the needs of the rest of the software. The second is to determine the exact size and orientation of the LPROM and maintain the correct orientation during the actual programming process. The final task for the software is to drive the stepper motors which control the X-Y translation stage beneath the laser and to correctly direct the laser energy to the memory location to be programmed on the LPROM. The "C" programming language was chosen for its ability to operate on different

data forms and it's ability to interface with the control software for the ITEX FG-100 boards and Modulynx controller.

4.3.1 **Software Interface.** To maintain compatibility with existing CAD tools, the automation software needs to be able to accept data generated by the *assem* microcode compiler which is the same format as data inputted into the XROM. Symbolic microcode is translated by the generic microcode assembler tool (GMAT) which then calls upon the *assem* compiler to correctly format this data. The output is a stream of unsigned numbers up to 32 bits in magnitude. For example, the largest possible number in an output stream of 12 bit numbers is 4096. To facilitate horizontal microcode formats, which are typically very wide, and adhere to the 32 bit limitation, the GMAT program automatically divides the total microcode word into four equal parts. To reconstruct the original desired microcode word, the output stream from GMAT/*assem* is divided into groups of four consecutive words which are then concatenated. Consequently, the widest complete microcode word that can be output by GMAT is 128 bits.

The data filename to be read into the program which contains the *assem* output is declared in a LPROM_PARAM.h file which is edited by the user prior the programming operation. In addition, the following data is also required to be declared: the width of each individual word, DATAWIDTH; and, the total number of words, NUM_WORDS.

This format of unsigned numbers; however, is not compatible with the laser programming requirement of the software which dictates that the numbers of the output data stream be converted into a bit stream of ones and zeros. To do so, after all the data is read into an input array, the elements of this array are grouped and examined four at a time. A mask is generated which contains a one in the most significant bit and zeros elsewhere. This mask is "anded" with the first of the four words after which this word undergoes a bitwise shift to the the left to place the next most significant bit in line with the masked bit. The result of the anding is stored in a preliminary array which will be further manipulated latter. Each of the four words receives the same treatment until all the bits of the microcode word have been individually examined and stored in a preliminary array.

Next, the LOCKCOL variable in the LPROM_PARAM.h file is checked. A LOCKCOL value of *"1"* implies that the order of the columns of the LPROM has been predetermined to permit the LPROM to be mated to an optimized XROM. Under this condition, both the XROM and LPROM will have their outputs multiplexed together on the same data bus and it is assumed that both memory devices will be addressed as one memory device. An xromlock.in file, which stores the XROM column ordering, must exist in the same directory as the LPROM_PARAM.h directory.

To facilitate XROM optimizations using wordsigns, the second and fourth subarrays of the XROM are reversed so that they may share wordsign cells with the first and third subarrays. Since, by definition, the microcode of the LPROM is unknown prior to fabrication, wordsign optimization is not practical; however, if the LPROM is to be mated to the XROM, it to must have its second and fourth subarrays reversed. Setting the XROM_CONFIG bit in the LPROM_PARAM.h file will ensure these LPROM subarrays are reversed.

The next step in data formatting is to account for the 2 TO 1 multiplexer at the base of the LPROM sense amplifier. This multiplexing allows only every other bit to be transmitted through to the data bus. Consequently, the bits of consecutive, complete, microcode words have to be alternated when forming a row of bits to be programmed into the LPROM. The bits of the microcode word with the least address precede those of the word with next least address. Figure 4.5 below illustrates the data formatting process from the raw *assem* data to the out_array.

4.3.2 **Pattern Recognition Software Operation.** The pattern recognition algorithm used can be divided into two parts. First a template must be generated using the image to be found. Second, once the circuit is correctly positioned beneath the objective, the computer must determine the location of the "taught" image on the screen. Several intermediate steps are added to the algorithms to decrease the total number of calculations need to perform the necessary correlations.

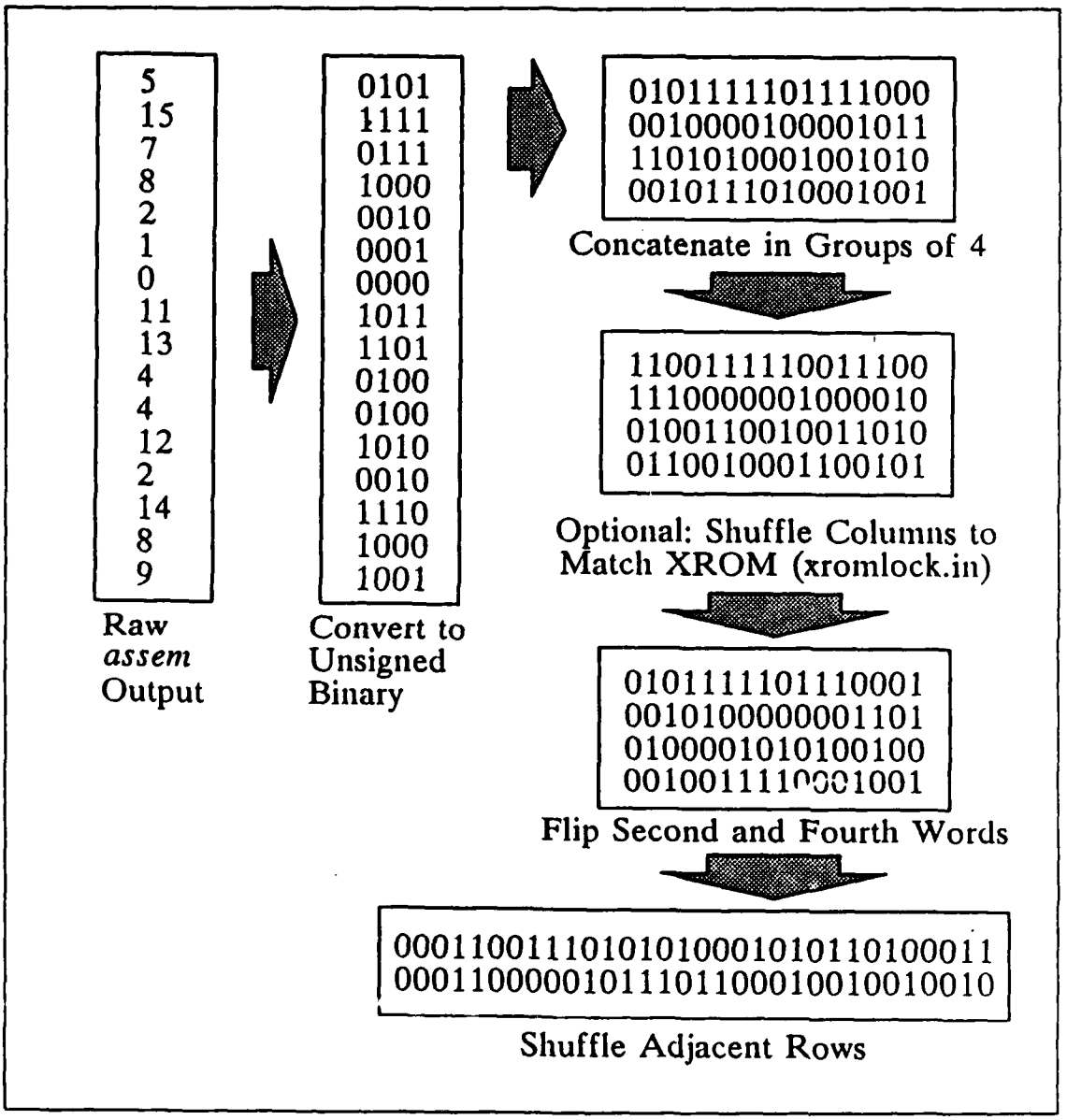


Figure 4.5. Data Formatting Process.

4.3.2.1 **Generating Pattern Recognition Templates.** Template generation begins by placing a cleaned LPROM circuit on the translation stage and viewing it microscopically through the video camera. Dust particles on the circuit will introduce errors in the correlation and increase the risk of having the software identify the wrong image. This video signal is also piped into the ITEX FG-100 image processing board where a single frame of video is grabbed, frozen on the screen and saved as a pixel array. To reduce the noise in the image, which contributes to fussiness of the picture, the image is passed through a low pass filter. This filter removes any sudden, isolated changes in pixel intensity. The next step is to reduce the size of the image from a 512 x 512 pixel array to a 256 x 256 array. This is done using the ITEX "squish" command which removes every other bit in both the X- and Y-directions. A minimal amount of image resolution is sacrificed to reduce the size of the pixel array. Since the correlation requires $3n^2$ x m operations, where n is the size of the template and m is the size of the area being scanned, this 4 TO 1 reduction in image size can result in a significant time savings. The pixel array should be saved at this point. With the aid of the on-screen graphics commands, the programmer determines the exact size and location of the alignment mark within the "squished" image. The feature in the template serving as the center point should be noted. The filename of the pixel array and the dimensions and location of the center of the alignment mark form the template and are all declared in the LPROM_PARAM.h file. This procedure needs to be performed to generate a template for each alignment mark.

4.3.2.2 **Determining Screen Coordinates of Template Image.** To determine the position of the template image on the video screen, a time based correlation between the saved template image and the present video image is performed. To maintain correct scaling with the template, the grabbed video image must also be squished. If the image is poor, it must also be passed through a low pass filter prior to squishing. Within the dictated scan limits, an area, identical in size to the template is compared to the template itself by means of a time domain correlation. In this correlation, the total energy of each image is normalized so that the sum of the energy of all the pixels in the entire array equals one. Next, the normalized energy of corresponding pixels in each array are multiplied together and their products totalled. This sum represent the correlation of

the two images. The template is then shifted by one pixel and the process repeated until the center of the template has scanned over each pixel within the dictated scan area. The software retains the X- and Y-coordinates of the pixel yielding the highest correlation. Figure 4.6 below illustrates both the template generation process and the scanning process of the template over the video image.

The scan area can be the entire grabbed video image or any subset of the image. The minimum scan area which ensures the image to be found will be scanned needs to be experimentally determined. The boundaries of the scan area are declared in the *limits*[] array in the *find_mark* subroutine.

Theoretically, this correlation algorithm is capable of resolving position down to one pixel on a 256 pixel wide squished image. A 60x objective will produce a 71 μm wide field of view. Thus, maximum accuracy of the pattern recognition software is $\frac{71}{256}\mu$m = .2776 μm. The actual accuracy will be discussed in Chapter 5.

4.3.3 **Scale and Fabrication Error Compensation.** The second requirement of the software is to determine the true size and orientation of the fabricated LPROM. And, once programming begins, the software must maintain correct alignment between the laser and the LPROM. Both of these procedures make extensive use of the pattern recognition software described above.

4.3.3.1 **Scale and Fabrication Error Determination Prior to Programming.** This software is necessary because of the variations in the actual size of features fabricated on the circuit. Some vendors may fabricate features smaller or larger than than the advertised scale factor to allow meet advertised performance specifications. The actual scale factor, lambda (λ), has to be determined precisely to avoid cumulative step errors, which may become significant over the span of a wide LPROM, due to an incorrect assumption of the fabrication scale factor. In addition, rotational error may be introduced if the die is not packaged squarely or the package is not mount squarely on the translation table. This error must also be determined and compensated. This information

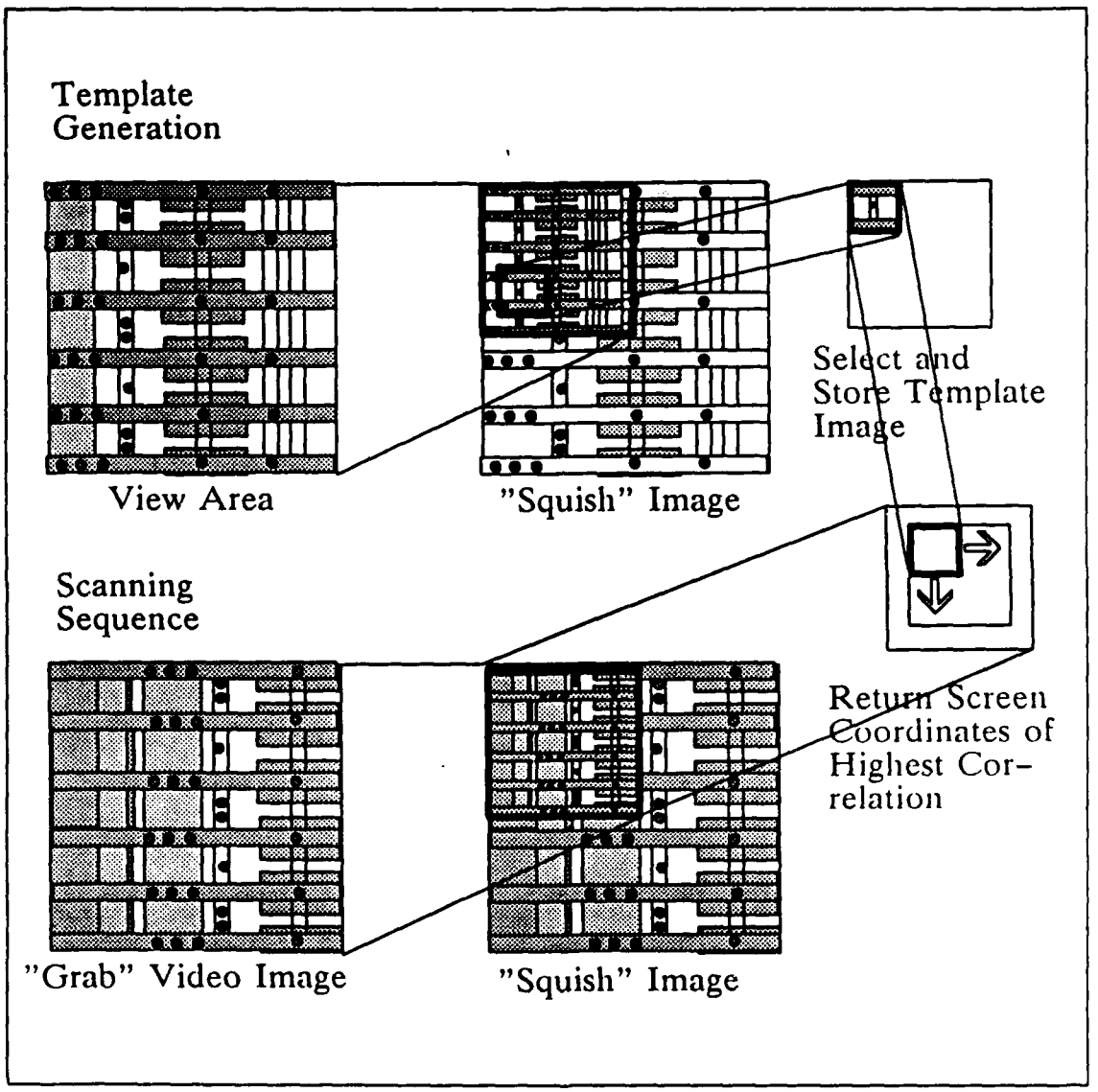


Figure 4.6. Pattern Recognition Correlation Process.

will be derived after using the pattern recognition software to locate the four corners of the LPROM.

The first step in determining the true scale factor is to have the programmer provide the theoretical LPROM dimensions in the LPROM_PARAM.h file. These dimensions include: the theoretical λ, FAB_TECH; the width of the LPROM, HLEN; the height of the LPROM, VLEN; and, the horizontal distance from the alignment mark to the first laser pulse site, INIT_STEP. Combining this information with the knowledge of how the LPROM is designed, the horizontal and vertical step distances between laser pulse sites are determined. This information is then used to make an initial guess as to the true location of the four corners of the LPROM.

The programmer is next required to position the feature designated as the center of the alignment mark in the middle of the screen, designated by the spot marker light. The software then resets the stepper motor position data in the MITAS controller. This is now considered the origin of the coordinate axis to be used in determining the size and orientation information.

Using the initial dimension information, the table will translate to the expected position of the lower right corner at which point the pattern recognition routine is called to verify the true position of the lower right alignment mark. The differences in X- and Y-coordinates are tabulated and then used to in determining the new expected location of the upper right corner of the LPROM. The table translates to this location, the pattern recognition software invoked, the true location determined and tabulated and the new expected coordinates of the next alignment mark calculated. The above process is repeated automatically for the last two corners. In using this algorithm to determine the size and position of the fabricated LPROM, I am assuming that the mechanical errors induced by the stepper motors are small. Errors that do exist will be averaged over the entire length and width of the LPROM.

The X-Y position data of each corner is then used to determine any rotational error. The results from each corner are then averaged to arrive at the rotational or orientation error, θ. The corner position data is also used to determine the true length of each side

of the LPROM. This information is combined with the theoretical length to generate a scale factor. Both the scale factor and the θ rotation are used in a translation filter which preprocesses coordinate translation information prior to passing it on to the stepper motors.

4.3.3.2 **Alignment Correction During Programming.** Once the true size and orientation of the LPROM are determined, X-Y table translation for laser programming becomes simply a matter of correctly aligning the laser over each laser link. The only way to ensure this is to provide position feedback before each laser blast. The Anarad Table, with its linear encoding, can provide this feedback quickly. Without this table, the only other available means of providing feedback is to use pattern recognition.

The Use of Pattern Recognition During Programming. It would be desirable to use pattern recognition prior to every laser blast. However, preliminary analysis of the MicroVaxII for the time necessary to recognize an entire LPROM memory cell and ensure correct alignment is one minute. Thus, in its present configuration, a 2K LPROM would require an unacceptable minimum time to program of a day and a half. In a compromise between duration of programming and necessary accuracy, I decided that pattern recognition would be used only at the beginning of each row and that steps need to be taken to minimize mechanical errors such as gear lash and vibrations.

A computer system capable of controlling both the ITEX FG-100 boards, the Modulynx stepper motor controller, and can perform the correlations faster, such as a Sun4 with array processor boards, may make pattern recognition before every laser blast practical.

Externally induced vibrations is a hardware problem that can be minimized by using an nitrogen-suspension, anti-vibration table. Removal of gear lash requires both hardware and software.

Gear Lash Compensation. Gear lash is defined as the error induced into the movement of the X-Y translation table due to an imperfect meshing of the teeth in the drive gears of the table whereby a rotation of the adjustment knob does not result in an immediate movement of the table. Fortunately, gear lash is not cumulative and occurs

only when the direction of rotation of the adjustment knob changes. If the gear lash proves large enough, it can be minimized by first compensating for it prior to reaching the alignment mark at the beginning of each row, and then, by maintaining the same direction of rotation of the adjustment knobs until the next alignment mark.

To account for the gear lash prior to reaching the alignment mark, both stepper motors are instructed to rotate away from the expected location of the mark and then return just short of this location. The minimum step size away from the mark will have to be experimentally determined to ensure all the gears have been engaged in both directions. Since the programming of the rows will always be from left to right, the step directions for the X-axis stepper motor will always be first left then right with the alignment mark residing in the right half of the screen. The step direction of the Y-axis motor will depend of the value of the sine of theta. A theta value of zero or greater implies that, as the table progresses to the right, the Y-axis motor will either remain stationary or rotate upwards. In this case, the initial step will be downwards first and then upwards with the alignment mark residing in the upper half of the screen. If $\theta \leq 0$, the Y-axis movement will be upwards first and the downwards with the alignment mark to be found in the lower half of the screen. Thus, considering both the X and Y-axis motors, if $\theta \geq 0$, the alignment should be found in the upper right hand quadrant of the screen; if $\theta \leq 0$, the lower right hand quadrant.

If the alignment mark is found in an incorrect quadrant, the stepper motors are instructed to step away even further before returning. The size of each motor's step will be dependant upon which quadrant the alignment mark was found. If the alignment mark is not found on the screen at all, the programmer is instructed to use the keyboard curser controls, which move one full screen in any one of four directions, to place the mark back on the screen after which the pattern recognition software once again searches for the mark and determines the correct movements for the stepper motors. This intentional offset to the alignment mark is fed to the main X-Y stage control program where it is included in the step distance calculations to the first memory location in the row.

4.3.4 **X-Y Stage Control.** After removing gear lash and determining the actual distance and direction to the first memory location, the Modulynx controller is fed the coordinates to this point and subsequently fed the coordinates of every memory location to receive a laser blast in the row. This controller returns a DONE signal to the calling routine. This will prevent the software from getting ahead of the hardware. Since all coordinates forwarded to the Modulynx controller must be given in terms of rotations or partial rotations of the stepper motors, the coordinates, which are first calculated in terms of microns, are then multiplied by an experimentally determined scale factor which is function of the controller's own scaling. The controller's parameters, which must be set prior to starting the automation software, are provided in Appendix B.

4.3.5 **Software Integration.** All the programming subroutines come together in a program called *LPROM_PROGRAM.c*. The only required software that is not executed as part of this program are the *make_temp* routines which are used to generate the patterns or "templates" to be learned for pattern recognition. This template need only be generated once for each new LPROM fabrication run. The software design is modular with each individual task being written as a separate subroutine. Figure 4.7 shows the various relationships between the different subroutines.

The first subroutine executed as part of the *main* program is a checklist of the required equipment that must be turned on, warmed up and calibrated; also the required LPROM parameters that must be input into LPROM_PARAM.h. The next routine formats the data into the *final_out* array. This is followed by the request for the alignment mark to be centered in the screen after which, the the scale and orientation software executes. The final software to be executed steps each memory location of the LPROM beneath the laser; checks the *final_out* array to determine if the site is to be programmed; and, calls a *pulse_it* subroutine to activate the shutter if it is. All the software generated as part of this thesis effort is contained in a limited distribution appendix to this report.

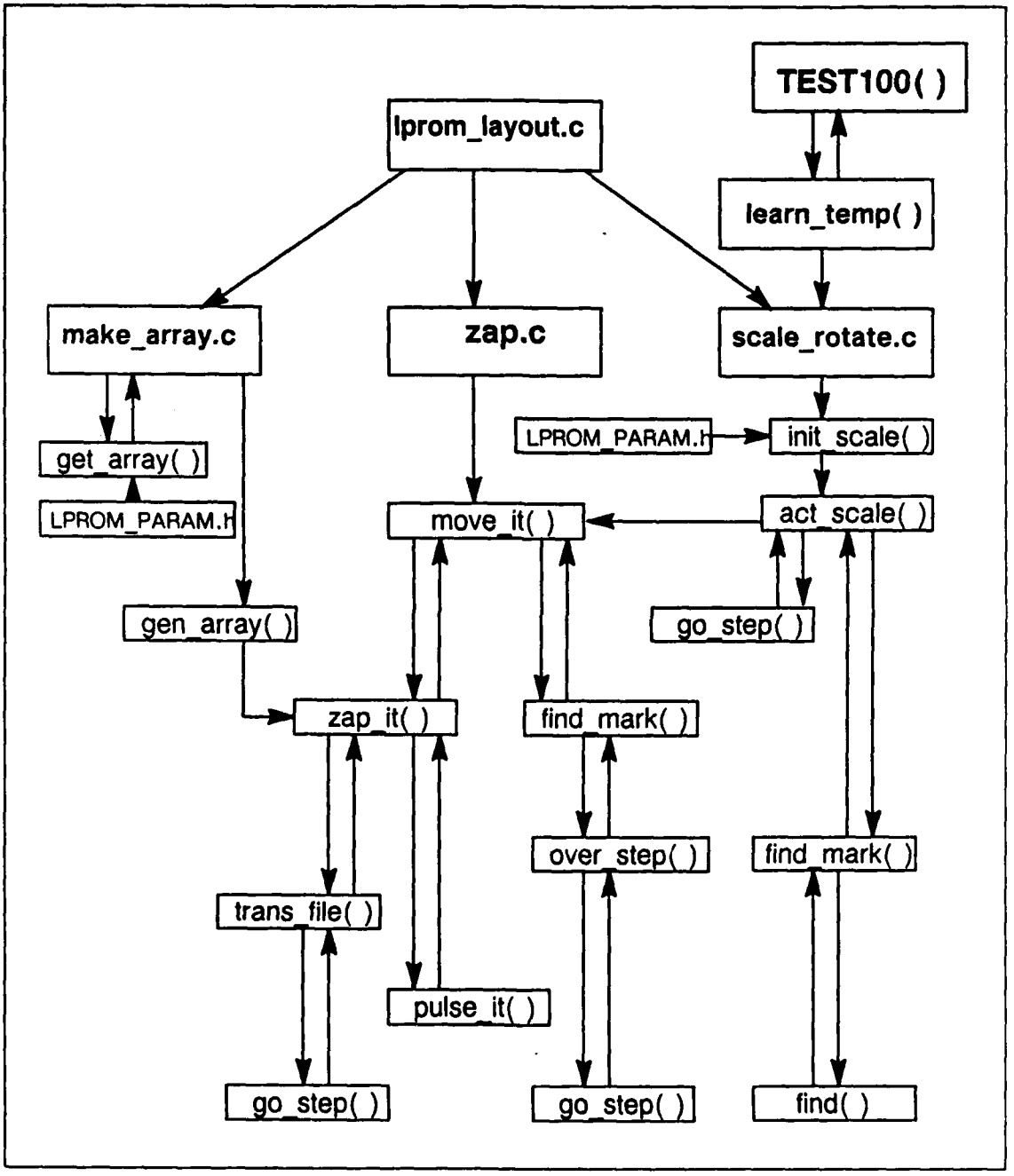


Figure 4.7. Software Subroutines.

4.4 System Integration

The final phase in the design of the laser programming station is to mate the software to the hardware. The main interfaces between the MicroVaxII and the optics box with its X-Y translation stage are the ITEX FG-100 image processing board and the MITAS controller. Both of these devices came complete with their own software. Interfacing became a matter of having the main program make the appropriate subroutine calls at the right time. The only additional interface requires running a wire from a parallel interface port on the MicroVaxII to the shutter controller. The *pulse_it* subroutine pulses this wire whenever a memory location is to be programmed. For synchronization reasons with the shutter timing circuitry, the pulse must be 20 to 30 μsec.

4.5 Summary

The Florod laser optics box alleviates some of the beam conditioning problems encountered by previous research efforts at AFIT. The two stage shutter with its synchronization hardware ensures the proper pulse length of laser energy is generated. The pattern recognition software ensures this energy is directed to the correct location on the circuit. The automation hardware and software eliminates the potential for human error as a result of manual programming.

V. Results

5.1 Introduction

This chapter discusses the results obtained during the testing of the various facets of this thesis effort. Since much of the research is dependent upon the correct fabrication and operation of the LPROM test circuit, this will be the first item studied. The operation of the laser programming station is next followed by the results of actually programming the test circuits. The final discussion of this chapter deals with the parameterization of the programmed LPROM.

5.2 Fabrication Results

Fabrication results include visually inspecting the circuit for fabrication errors such as excessively bloated or shorted metallizations and powering up the unprogrammed circuit and ensuring it functions correctly.

5.2.1 **Visual Inspection.** Upon receipt of the LPROM test circuits, I compared the CIF plot of the circuit with the pin-out description provided by MOSIS. It became immediately apparent that there was a discrepancy between what I thought was pad #1 on the circuit and what MOSIS labelled pad #1. Their pin-out description has pad #1 two positions further down on the right side of the circuit which places it on the LSB data output pad rather than a ground (GND) power pad. Since pin #1 is connected to the substrate, it is important that it's attached to ground to prevent placing all the *n*-type transistor in a virtual floating well.

Examination of the circuit under the microscope revealed that neither description was correct and that pin #1 is actually only one position down from my original understanding of the TinyChip pin-out. While this pad is still an output pad, there is no output signal connected to it. The substrate can therefore by tied to ground through this pin. There are redundant GND pads which ensure sufficient power for the circuit. To prevent any potential fighting between the input GND to this pin and the pad itself, the bonding wire was removed.

Another visual flaw is the over bloating and shift upward of the horizontal *metal2* lines. This defect affects all the *metal2* lines, but most significantly, the horizontal ground lines and the upper *wordline* of a *wordline* pair which connect the diffusion wells to GND. The bloated *metal2* covers the *n*-diffusion in the cells located above these lines. This will create difficulties in programming all of the memory cells since the diffusion may not receive the direct laser energy it needs to heat and diffuse the charge carriers. The extent of bloating by each vendor is not available from MOSIS. Consequently, in the future, new LPROM memory cells should allow more area for this potential error. Figure 5.1 below is a photograph of these malformed cells.

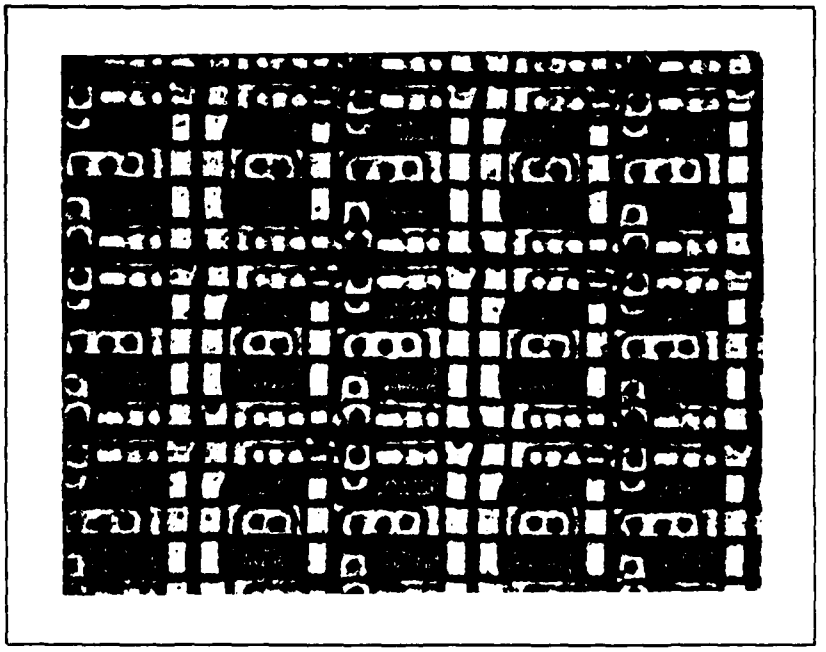


Figure 5.1. Photograph of LPROM Memory Cell.

5.2.2 **Power Up.** Initial power up is performed with 100Ω resistors in series with all power input pads. In the event of latch up, these resistors will limit the current through the circuit to 50 milliamps. An ammeter is placed in series between these resistors and the power supply and an oscilloscope monitors the output voltage from the power supply.

This voltage is slowly increased and the ammeter monitored for any sudden increases in current which could signal the onset of latch up.

Power up under these condition revealed a perfectly linear relationship between current and voltage. The resultant I-V curve equated to a resistance of only 3.2Ω.

Visual inspection failed to reveal any fabrication flaws. Analysis of the CIF plot; however, did reveal two shorted power busses. A location was identified where these two *metal2* lines should be separated. One *metal2* line could be easily scribed with a picoprobe under the Micro-Manipulator probe station. The other location is directly adjacent to two vias and cannot be scribed without risk of irreparable damage to these vias. The only alternative was to use the laser programming station to cut the *metal2* lines before using it to program. This procedure is discussed below. An additional design error required a picoprobe to supply GND to the AND plane address decoders.

After the short was removed with the laser, the Vdd to GND resistance rose significantly. Power up revealed only a 4.5 mA current through the circuit at 5V. Application of a 1 Mhz, 50% duty cycle, 5V pulse train to the precharge pad increased the current through the circuit to 45 mA. There was no evidence of latch-up.

5.2.3 **Memory Content Verification.** Once the circuit was powered up, the address lines were alternately tied to Vdd and GND to generate the input addresses while a probe monitored the output data. As expected the signal at the output pads did not deviate from 0V indicating the contents of the LPROM array to be zero.

5.3 Programming Station Accuracy

The small size and numerous memory locations of the LPROM array emphasize the importance of correct laser blast placement. This section discusses the results of testing the movement accuracy of the stepper motors, the accuracy of the pattern recognition software and the results of combining these two processes.

5.3.1 **Stepper Motor Movement Accuracy.** Two initial tests were conducted to determine the accuracy of the stepper motors without the use of the pattern recognition

software. In the first test, the same set of coordinates used by Capt. Spanburg in his position accuracy tests were fed to the controller. After traversing 713 μm the controller was instructed to return to its origin point. The results of this test were encouraging. At the conclusion of each run, if the spot marker missed the origin, it did so low and to the left and by never more than 2 μm. This is an order of magnitude better than belt driven system.

In the second test, the spot marker was once again illuminated on the first alignment mark and the stepper motor controller fed the coordinates of the subsequent memory locations. One fact became clear rather rapidly. The movement between the LPROM memory cells across the array was *not* consistent. The movement made between cells on the left side of the LPROM tended to be first too large, resulting in a movement of the spot to the right side of individual LPROM memory cell. The steps are then apparently shortened resulting in a drift of the marker to the left side of the memory cell. By the time the movements have reached the right hand side of the array, the spot marker is once again on the right side of the memory cells. The problem is severe enough that it became impossible to position the spot marker at any given point at the beginning of the array and guarantee that the laser would not contact either a via or other metallization somewhere along the row.

The results were consistent across a multiplicity of rows and on different chips. Examination of the array under an optical microscope failed to reveal the apparent 7 μm difference in memory cell size inferred by the right-left-right drift. The software was modified to display the exact number of steps passed to the controller prior to each movement. The step size was consistently 182 with every sixth step being 183. This is due to round off error in the casting of the floating point, actual distance value into an integer number of steps which is the required input to the Modulynx controller. This round off is insufficient to account for the drift.

These results point the finger of blame at the stepper motors and controllers. There is no feed back between the controller and the motors. The controller sends out a series of pulses to the different poles of the stepper motor. It has no way of detecting if the motor has successfully rotated. Further examination of the problem reveals that there is a high

correlation between the direction and magnitude of the position error and the rotation position of the stepper motor. The entire width of the array is 832 μm. One full rotation of the stepper motor translates into a 635 μm movement.

In an attempt to confirm this apparent sinusoidal error, a test was performed in which the LPROM was placed on the left, right and center of the X-Y translation stage. The initial left hand side alignment mark was centered in the video screen. The stepper motors were then instructed to step to the individual memory locations. Using the left edge of the diffusion well as a reference, the position of the memory cell, relative to the first memory cell's location, was recorded. This process was then repeated with the X- and Y- stepper motors switched. In addition, the LPROM was placed in the center of the stage and the X-translation stepper motor rotated manually. Assuming a 26 μm step size, the micrometer was rotated 10.25 mils for each step. The results of each of these 7 tests were plotted and are included in Appendix E of this report.

These results were both surprising and disappointing. There is no discernable pattern to the error. The earlier observation of a sinusoidal error function was a coincidence. The apparent repeatability of the error was due to the use of a zero insertion force (ZIF) socket mounted to the MS-166 stage. The ZIF socket would hold the chip in the same Y-plane with just minor X-plane variations. In addition, the near perfect first position tests always concluded with a return back to the origin which places the stepper motors in the same orientation and apparently cancels out this error.

Still, if the error is truly repeatable from row to row, the error could be characterized by a single pass along a row of memory cells. The offset for each column could then be included as part of the step size calculations performed during laser programming. Using the same test as before, the position errors were recorded for four consecutive rows. The plots of these results are also provided in Appendix E. Since pattern recognition was not used in this test, any realignment error at the beginning of a new row was recorded and accounted for in generating this data.

As these plots illustrate, the error from row to row is very similar but not exact. Superimposing all four plots on the same axis reveals row to row deviations as high as 2

μm. This error is right at the limit for damage to surrounding VLSI structures established by the actual programming results. The consistent rise in error from left to right can be corrected by a change in the step size scale factors.

Another possible solution is the integration of pattern recognition prior to each laser blast. However, as previously mentioned, this is very time consuming and may be stretching the limits of practicality without some improvement in image processing time, either with faster hardware of developing a faster algorithm. Two versions of the automation software have been generated. One with a stepper motor error characterization run prior to programming and one with pattern recognition integrated into every laser zap. This second solution obviates the alignment and orientation, and the anti-gear lash software.

An obvious hardware solution is to replace the stepper motors with higher precision units.

5.3.2 **Pattern Recognition Accuracy.** Initial tests of the pattern recognition software revealed several flaws and malfunctions. Prior to reporting the results of the alignment accuracy tests, the events leading to the determination of the source of the errors and their correction are discussed.

5.3.2.1 **Corrections to the Pattern Recognition Software Package.** The initial tests of the pattern recognition software described in Chapter 4 used a diffusion well contact in line with rows of LPROM memory cells as the center of the pattern recognition template. Without moving the circuit, the software was then tasked to search the same screen using the new template and return the coordinates of the pattern.

The initial results were very disappointing. The returned coordinates seemed almost random and the correlations were all under 0.3, which is practically no correlation at all. Unfortunately, the ITEX FG-100 video boards have a history of performing incorrectly under low light conditions.

To increase the brightness of the image to the video camera, the 6V illumination lamp on the optics box was replaced with a 12V light (an automotive backup light works well). In addition, there is an orange colored optical filter designed to protect the video camera

and microscope observer from laser light reflected off the internal optics and the circuit itself. This filter is epoxied into place on a mirror and prism assembly and contributes to the lack of light reaching the camera. Upon contacting them, Florod agreed to forward a prism assembly without the filter in place and include additional filters to be placed in the eye pieces to continue to protect the microscope observe from retinal burns.

The same tests with the 12V bulb yielded higher correlations but no greater accuracy. Correlations were consistently above 0.85 but would find the correct image less than 50% of the time. Modifying the user defined parameters (*scan size*, *template size*) varied the average correlation, but the highest values were still being located incorrectly.

After correcting flaws in the pattern recognition software package failed to improve the system's accuracy, the entire correlation routine was rewritten using new variable names, removing the ITEX supplied TEST-100 "squish" command and removing the some of the optimization algorithms. This new software functioned correctly. The errors were then traced back to the "squish" command and one of the optimization routines. The "squish" apparently processed the pixel data beyond simply removing every other pixel in the X- and Y-directions. The erroneous optimization routine was indicating each new correlation to be higher than the previous one regardless of the actual calculations.

Still desiring a reduced template and a reduced grabbed video image, a new "squish" routine was written which averages the values of four adjacent pixels from the full size image and places the value in a smaller array. This new routine functions correctly with the rest of the correlation software and has the added benefit of reducing the effective noise in the image thus obviating the use of the low pass filter.

5.3.2.2 **Pattern Recognition Accuracy Results.** Using the smaller sized template and grabbed video image, the template size was gradually decreased until the software failed to report the desired image as the highest correlation. Thus, for the four corner alignment marks, the minimum template size was experimentally determined to be 30 x 30 pixels. Under this condition, the software consistently found the location of the desired image to within one pixel. For the stepper motor error characterization software, a much large template of 75 x 75 pixels is required to ensure correct identification of the entire

LPROM memory location. This high degree of accuracy can be supported by considering the large number of pixels being examined in the correlation process. Figure 5.2 below is a 3-dimensional plot of the correlation of a typical 30 x 30 template image as a function of the template's X- and Y-position on the grabbed video image.

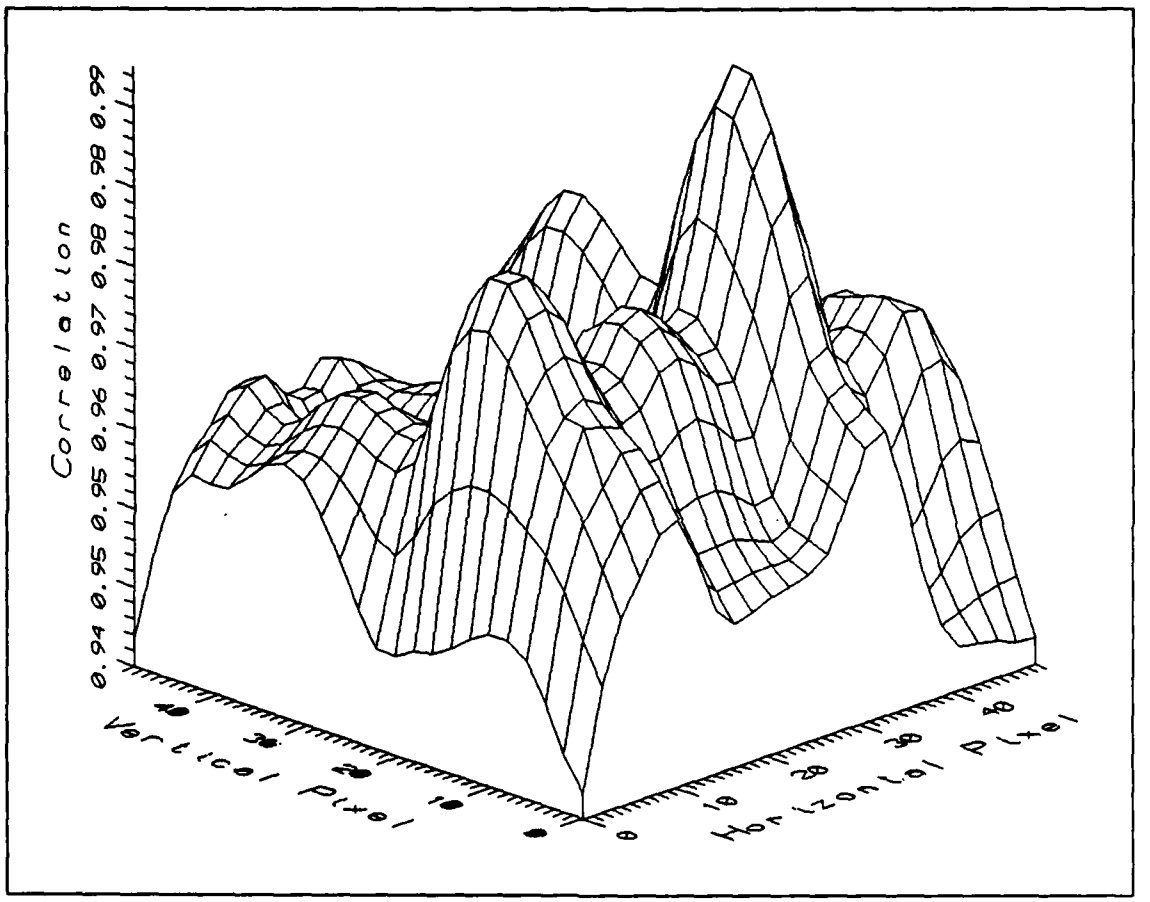


Figure 5.2. Plot of Template and Video Image Correlation as a Function of X- and Y-Position

The minimum scan area is a function of the accuracy of the stepper motors. Assuming a worst case deviation of 8 μm, as derived from the stepper motor error characterization tests, the minimum scan area is 60 x 60 pixels and 95 x 95 pixels for the 30 x 30 and 75 x 75 pixel templates respectively.

An important observation is that, to function properly, the software assumes that both the template and grabbed image have identical orientations. Any deviations result in a significant reduction in the peak correlation. Consequently, the user is required to reduce the orientation error as much as possible prior to run the pattern recognition software.

5.3.2.3 **Actual Size and Orientation Accuracy.** The software for determining these parameters is only as accurate as the offset coordinates it receives from the pattern recognition software and the accuracy of the stepper motors. The premise of this software is that the stepper motors are accurate. This is obviously not the case. Thus, the worst case error can be calculated by adding the maximum measured stepper motors error of 8 μm and the 0.2776 μm error of the pattern recognition software and dividing by the total distance translated between the alignment points, in this case, 831 μm to yield 0.996%. In other words, for an LPROM designed to be 831 μm wide, the true width should still be measured within almost 1% even with the larger than expected stepper motor error. The total height dimension accuracy is 1.6% and the θ accuracy is approximately 5.1%.

5.3.2.4 **Laser Pulse Position and Size Accuracy.** Like the actual size and orientation determinations, laser position accuracy is a function of stepper motor accuracy and pattern recognition error. As mentioned earlier, with the stepper motor error characterization software in place, maximum placement error approaches 2 μm. Pattern recognition is still required at the beginning of each row.

Fortunately, due to the high precision of the stepper motor to micrometer head couplings, gear lash is negligible and the existing gear lash compensation software of marginal value when compared to the other known errors.

5.4 Operational Speed

The operational speed of this laser programming station is a function of the time required to determine the actual size and orientation of the LPROM array, plus the time to traverse through one row of the array and determine stepper motor error, plus the time required to step through each of the memory locations to be programmed. The speed of

the pattern recognition software is extremely slow requiring over 2.5 minutes to perform each correlation even with the optimization routines in place. This configuration of the stepper motors trades accuracy for speed.

5.4.1 **Stepper Motor Movement Speed.** The speed of the stepper motors is contingent upon the amount of accuracy desired. The automation software calculates a given step size. The Modulynx controller applies its own scale factor to the step size and moves to the desired coordinates. Increasing the step size and reducing the number of steps will increase the rate the stepper motors will step to the next set of coordinates, but reduce the accuracy. In addition, the BASE speed, ACCELERATION, DECELERATION and HIGH speed can be set in the controller though these variables seemed to have less of an effect on the speed of movement. The exact settings and operation of the Modulynx controller can be found in Appendix B.

The fastest rate I was able to have the stepper motors move was 12 memory locations per minute. This disappointingly slow rate can be attributed to the firmware interface between the MicroVaxII and the Modulynx controller and to the slow rotation speed of the motors while under control. With the subroutine call to the controller commented out of the software, the computer generated and displayed the coordinates on the screen very rapidly. This indicates that the bottle neck is not in the calculation of the coordinates. It is possible to generate all the coordinates to be programmed ahead of time and just feed them directly to the controller all at once. However, this does not permit any feed back to the computer which still has to control the laser; thus, this is not a viable alternative. When compared with the rate of movement of the Micro-Manipulator system, the MS-106 stage requires 14 times more rotation to move the stage the same distance. This results in a slower but more accurate movement.

5.4.2 **Pattern Recognition Alignment Time.** The primary concern of this software is that it must find the correct alignment mark 100% of the time. If necessary, programming speed will be sacrificed to meet this goal. As mentioned earlier, the software will run faster if the search area is reduced. In addition to using the four pixel averaging

"squish" routines on both the template and the grabbed video image, and minimizing the scanned area, an optimization algorithm was added to the pattern recognition software.

Instead of performing a full correlation on all the pixels in both arrays for every pixel in the scanned area, a subroutine was generated that correlates only the center two most rows and columns of the template and grabbed video image. A full correlation of all the pixels is performed only if the resulting correlation of this cross is higher than any previous correlation.

In comparison, a full size 512 x 512 , low passed pixel array, using a 60 x 60 template on a 100 x 100 scan area requires 6 min. 34 sec. to correlate. The same imaged squished to 256 x 256 pixels, with a 50 x 50 scan area, requiring no low pass filter, and using a 30 x 30 template, requires only 2 min. 27 sec. to correlate. The 75 x 75 pixel template on a 95 x 95 pixel scan area requires 7 min. 07 sec. The use of the cross optimization only saved 20 seconds on the 30 x 30 template and just over 1 minute on the 75 x 75 template.

A description of the operation of the ITEX FG-100 and TEST-100 software and the procedures for generating pattern recognition templates is included in Appendix C.

5.4.3 **Actual Size and Orientation Calculation Time.** The time required to perform this operation is the sum of the time to run the pattern recognition software four times, once for each corner of the LPROM and the actual time spent moving from corner to corner. To this sum should also be added the time to perform the stepper motor error characterization. In this instance, the total time for the preprogramming calculations is 31 x 7 min. 07 sec + 4 x 2 min. 27 sec. + 6 min. translation time = 3 hrs. 50 min. 25 sec. Actual programming time is a function of how many bits have to be programmed. The motors only step to those locations that are to be programmed. However, an additional 30 x 30 pattern recognition will still have to be performed for each row. This adds an additional 2 hrs. 35 min. and 54 secs. Assuming a worst case of programming all the bits, the additional time to perform this operation is approximately 4 hours. Thus total time to program the entire circuit is approximately 11 hours.

5.5 Laser Programming Station Configuration.

Several deviations from the original laser programming station design were necessary to meet the minimum power and spot size requirements for laser programming. This section outlines the causes and results of these changes. The exact procedure followed in setting up and aligning the laser and optics box are included in Appendix D.

5.5.1 **Initial Laser Programming Station Configuration Results.** A Spectra Physics 2020 argon-ion laser was not available for this thesis effort. A Spectra Physics model 164 was used instead. Initially, the maximum output power, as measured by a Coherent power meter, was 3.8 watts. The initial laser programming station configuration is diagrammed in Figure 5.3 below. After traversing the two collimation lenses and the four mirrors, the power entering the optics box was measured at 2.9 watts. The two mirrors comprising the elevation adjustment fixture, or periscope, and the two collimation lenses are not tuned for the blue-green laser frequency and account for most of the power dissipated prior to reaching the optics box.

With the spot marker light on, the X-Y apertures were adjusted to generate a 4 μm x 4 μm spot size. Next, one of the four objectives was removed and a power meter placed beneath this opening. The power measured at this point was less than $\frac{1}{10}$ of a watt! Most of the energy was being dissipated on the sides of the X-Y aperture blocks. A discussion with Lincoln Laboratories revealed that they do not use the X-Y aperture block as previously believed. Instead, they use a two lens telescope system to focus the beam down to the correct size. The beam shape is always round under this arrangement but most of the power is transmitted through the optics to the circuit.

Fortunately, in addition to the X-Y aperture, the Florod LFA optics box also came equipped with a LRY telescope optics system. After installing these optics, the maximum measured power out the open objective increased to 2.4 watts. However, the focal points of the laser and the visual optics no longer coincided. The laser beam was focused best when the circuit appeared slightly blurred through the eye pieces. Acting upon the advice of the optics engineers at Florod, the telescope lenses were removed altogether. Other than the internal dielectric mirrors, the only other optics the beam encounters prior to reaching

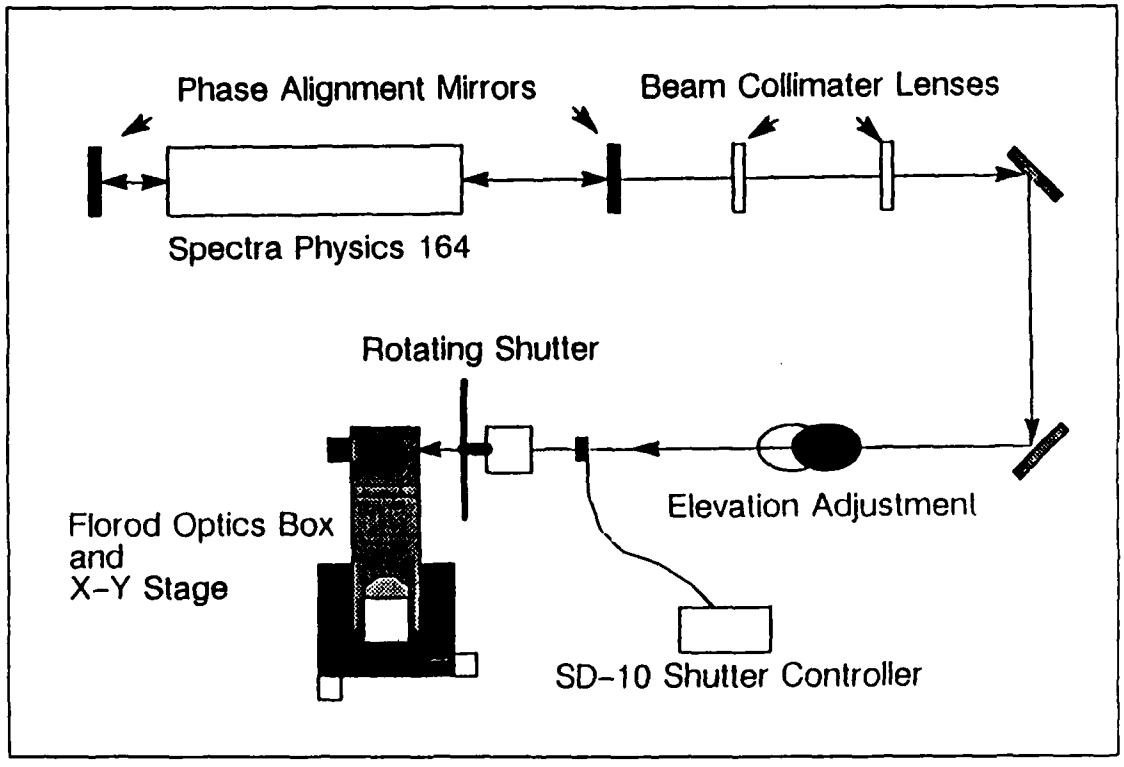


Figure 5.3. Initial Laser Configuration.

the circuit is the objective. The results were two fold. First, the output power measured at the objective increased to 2.85 watts, due mainly to the removal of the telescopic lenses which also were not tuned to the blue-green light of the laser. Second, the focus of the laser coincided better with the visual focus. Despite these improvements, this configuration still delivered an unacceptable level of laser power to the circuit.

5.5.2 **Final Laser Programming Station Configuration.** *To regain the necessary* power, a means of circumventing the metal front surface elevation adjustment mirrors had to be designed. An additional pair of dielectric mirrors and some creative alignment generated the necessary elevation change with out the metal mirrored periscope. The optic box remained free of lenses except for the final objective. This change coupled with a retuning of the laser resulted in a measured 4.6 watt output at the laser and a 3.9 watt measured power output through the objective. This new configuration, illustrated in Figure 5.4 below, provided the necessary power to proceed with the programming experimentation.

The rotating disk which forms the second stage of the two stage shutter was originally powered by a 3600 rpm Teletype motor. After this motor proved unreliable and subject to permanent thermal shutdown, it was replaced with a larger 3350 rpm motor. This slower motor increased the pulse durations by 7%. This change is reflected in the parameterization data below.

Despite the best efforts of the fabrication laboratory, the shutter and motor remained less than perfectly balanced. The vibrations induced into the air suspension table by the shutter were severe enough to prevent proper focusing of the optics. This problem was resolved by inverting the second stage and clamping it to an overhead shelf. The only difficulty with this configuration is the fact that the shutter and laser optics are now independent of each other. Changes in weight distribution on the suspension table, as a result of configuration changes, change the elevation of the table and require a realignment of the second stage of the shutter with the laser beam. This final shutter configuration functioned correctly delivering four different, slightly elongated pulse lengths to the optics box.

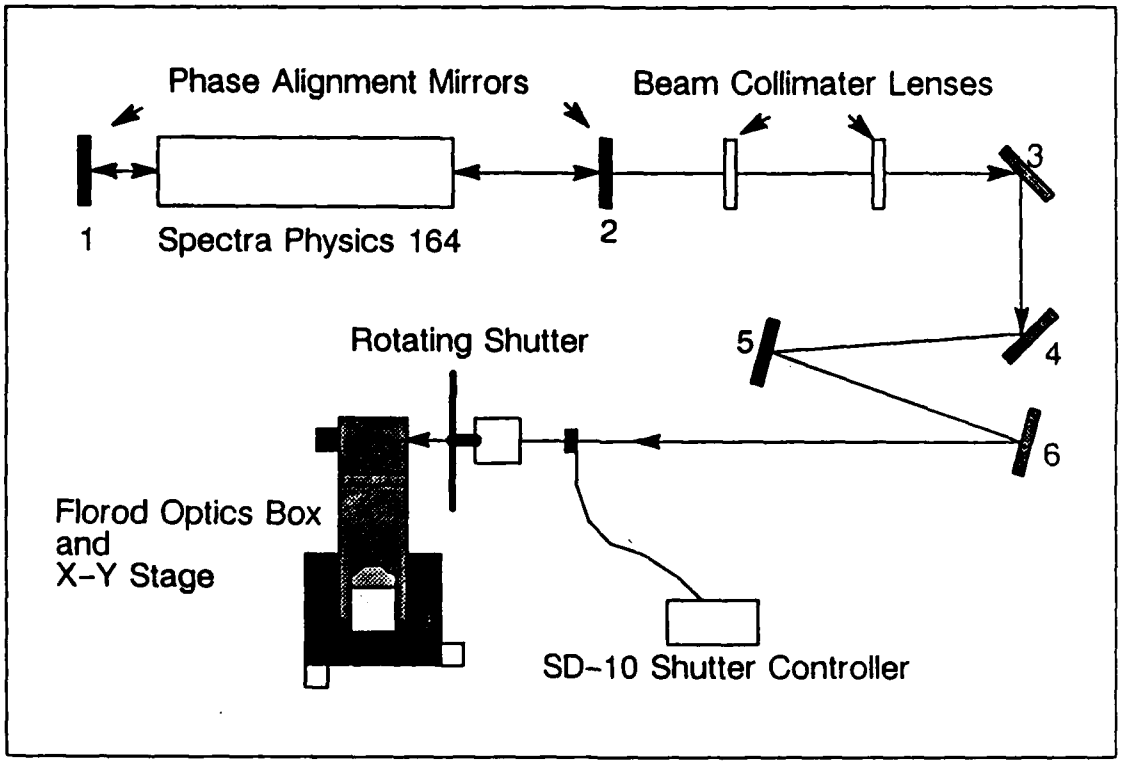


Figure 5.4. Final Laser Configuration.

Under maximum power conditions, the laser spot size is controlled by focussing and defocussing the objective. If less power and a smaller spot size is desired, the LRY's iris can be constricted to reduce the total amount of light reaching the objective. Since the laser light cannot be seen through the orange filter, spot size is experimentally determined by blasting holes through thermally sensitive material such as carbon paper. One discovery of note, while trying to determine the minimum spot size, is that minor variations in focus of the objective result in very large changes in spot size. Vertical movement of the objectives 2 μm away from the minimum spot size results in an increase in spot size from approximately 3 μm to 4 μm. This makes an auto focussing feature highly desirable for maintaining consistent spot sizes.

5.6 Using the Laser to Cut Metallization

While not its initial purpose, the laser programming station is capable of cutting both *metal1* and *metal2* metallizations. Using discarded integrated circuits as test fodder, laser and optics parameters were derived for cutting the metal lines. In principal, the metal absorbs the laser energy until it vaporizes at which point its rapid expansion causes it to erupt through the overglass [Yam85]. To avoid damage to underlying layers, pulse durations should be long enough to permit the laser energy to be absorbed by only the aluminum. Pulsed lasers, such as xenon and Nd:YAG, have short pulse lengths (submicrosecond durations), very high peak power (on the order of several hundred watts) and are ideally suited and commonly used for laser cutting. Using a much lower peak power, continuous wave argon-ion laser to cut posed some special problems. Once developed, these parameters were used to program a 32-bit comparator circuit and to correct the short circuit errors in the LPROM itself.

5.6.1 Metallization Cutting Laser Parameters.

Development of the parameters for laser cutting was divided into learning how to cut the two different layers of metal. *Metal1* is half as thick but twice as deep into the protective overglass as *metal2*.

For cutting lines away from active devices such as transistors or diodes, a 2 ms pulse at 4 watts (measured through an open objective) projected through a 40X objective proved

sufficient to remove minimum width, 4 μm, *metal2* lines. The larger spot size of the 40X objective permitted the removal of the entire *metal2* line with a single pulse. The 60X objective required 2 partially overlapping pulses to ensure complete removal of the metal. Even with the larger spot size, a minimum width *metal1* line sometimes required two pulses before it vaporized through the two layers of passivation. In general, wider metal lines have to be "nibbled" away with partially overlapping pulses. On some of the wider power busses, the metal is better able to dissipate the generated heat and requires several pulses before a break in the passivation oxide would occur. However, once the oxide was broken, the line cut rapidly.

For cuts that are near transistors or diodes where excessive heat can disrupt the electrical characteristics of the silicon, several short pulses are used. Three 96 μsec pulses at 4 watts is sufficient to nibble through a 2 μm wide *metal2* line. Despite the precautions of using a shorter pulse length, damage to the substrate is inevitable. Circuits designed in the future to be programmed by this laser programming station should take this into account and place the program metallization away from operational devices such as transistors and diodes.

5.6.2 **Programming the 32-bit Laser Programmable Comparator.** The 32-bit laser programmable comparator programmed and tested as part of this thesis effort is based on a similar circuit engineered by Capt. Spanburg and discussed in Chapter 2. The higher precision of the new laser programming station permitted the metal programming lines to be reduced in size considerably from Capt. Spanburg's original design resulting in a much more dense design. Figure 5.5 below is a CIF plot of one of the programmable cells of this circuit.

The cell will compare an incoming bit on the metal line labelled "a". Its inverse must be formed prior to reaching this cell and placed on the metal line labelled "abar". Cutting the line labelled "E" will disable the cell. This is necessary when less than the full 32-bits are being compared. The line labelled "P" controls how the cell reacts to incoming data on lines "a" and "abar". Table 5.1 provides a truth table for these comparator cells.

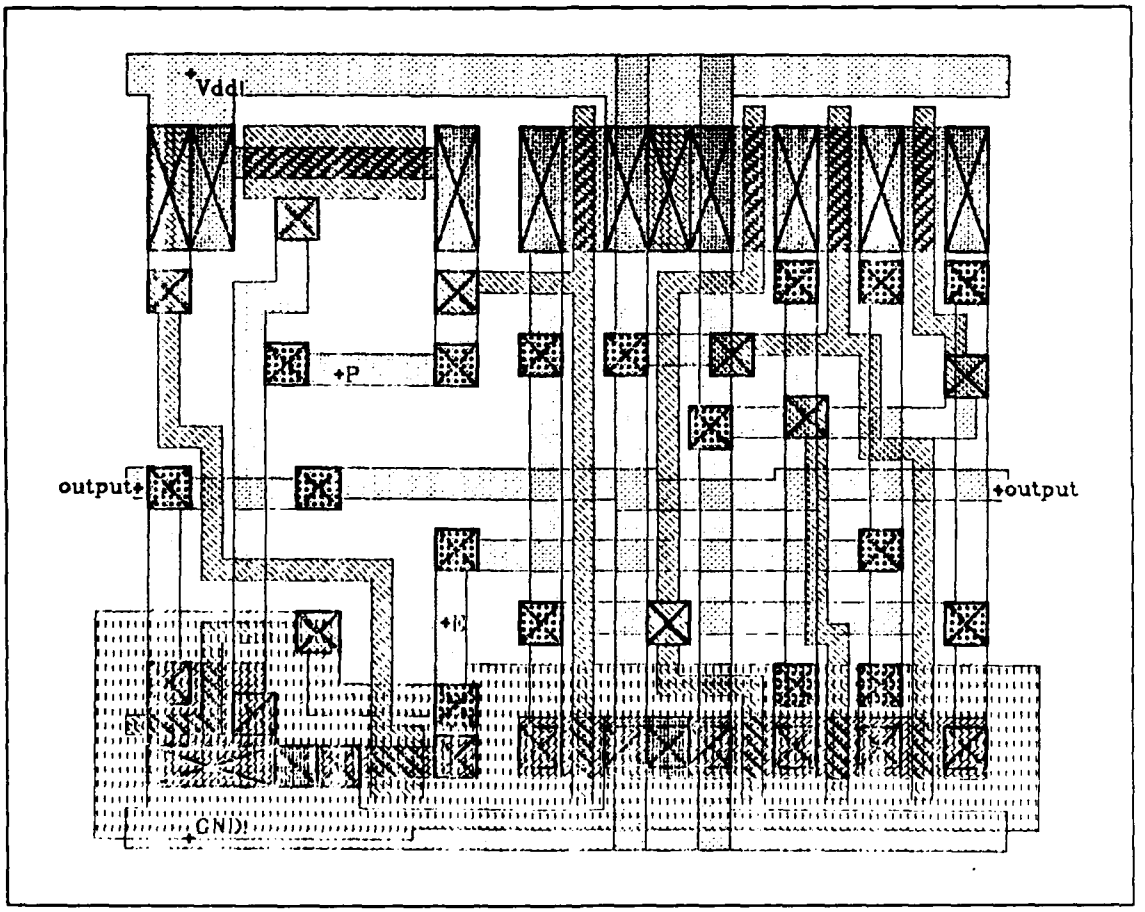


Figure 5.5. CIF Plot of Laser Programmable Comparator Cell.

E	P	a	abar	*Output*
C	X	X	X	X
U	U	1	0	1
U	U	0	1	0
U	C	1	0	0
U	C	0	1	1

C = Metal line is cut.
U = Metal line is uncut.
X = Don't Care.
1 = 5V(high).
0 = 0V(LOW).

Table 5.1. Truth Table for Laser Programmable Comparator Cell.

The metallization programming lines were fabricated in *metal2* and required three accurately placed 96 μsec pulses to cut. The "E" program metallization is positioned dangerously close to a polysilicon line and should be repositioned in future circuits.

5.6.3 **Using the Laser to Remove Short Circuits.** Two short circuits were errantly designed into the LPROM Tiny Chip. Both of them were approximately 10 μm wide and fabricated in *metal2*. They were easily and accurately excised using the laser with a 60X objective. *Metal1* to *metal2* vias placed close to the cut required a smaller cutting diameter; hence, the 60X objective. Earlier attempts to use picoprobes to scribe away the shorted lines succeeded only in mutilating these lines. The laser was also helpful in disrupting the surface oxide near probe points where the overglass was still in place. Once again, this should only be done away from active silicon devices.

5.7 Laser Programming Results

This section examines the correctness of programming for several different combinations of laser power and pulse duration. Aperture size was originally another parameter, however, due to the optics and power problems mentioned earlier, varying spot size was not possible.

5.7.1 **Effects of Varying Laser Power and Pulse Duration.** Due to equipment scheduling restrictions, the laser programming had to be accomplished separate from the automation effort. The circuit was manually positioned beneath the laser and the shutter was manually triggered.

Time restriction permitted only two pulse lengths and two power settings to be attempted. Both 96 μsec and 87 μsec pulses were used at 3.9 watts and 3.5 watts, measured through the objective. Figures 5.6 and 5.7 below are scanning electron microscope (SEM) photographs of programmed memory locations.

Figure 5.6. SEM Photograph #1 of Programmed Memory Location.

These photographs reveal several interesting points. Most importantly, they show the characteristic doughnut shape of a programmed diffusion link similar to the Lincoln Laboratories programmed links illustrated earlier in Chapter 2. The Lincoln Laboratories links did not have an overglass layer.

Figure 5.7. SEM Photograph #2 of Programmed Memory Location.

A second item of note is the cut made into the *metal2* line at the top of Figure 5.6 showing the result of laser energy coming in contact with the metallization. It is unknown how the diffusion link is affected by this incidental contact.

Also in these figures, the close proximity of the diffusion point to the polysilicon transistor gate can be seen. Serious consideration should be given to whether electrical characteristics of the silicon forming the channel have been disrupted by the forming of the link.

Interestingly, all four combinations of power and pulse duration resulted in similar looking links. Unfortunately, all attempts to exercise the LPROM and read a "1" from these programmed memory locations failed. The need for probe pads and test points became painfully clear as numerous attempts were made to probe both the *wordlines* and the *bitlines*. Since contact with these lines could not be definitively established, the probing results are inconclusive; though, none of the probe signals contradict the correct operation of this circuit.

5.8 Summary

The realities of performing research with limited resources resulted in modifications to many of the original plans and designs. Automation was established demonstrating the integration of feedback, through pattern recognition, with the stepper motor control software. Laser cutting of metal lines was established and the laser link parameters were visually validated. The surprisingly poor step size consistency presents special error correction problems. All proposed software solutions to this problem result in significantly greater programming time.

VI. Conclusions and Recommendations

6.1 Introduction

This chapter serves to analyze and interpret the results of the previous chapter as they apply to the solution of the problem statement. Shortcomings and recommended improvements in hardware, software, and VLSI design identified during this research effort are also presented in this chapter. Finally, follow-on research efforts are identified.

6.2 Conclusions

The goal of this thesis effort was to design an LPROM capable of being integrated with the AFIT XROM and engineer an automated programming station. The conclusions drawn from the results of the research efforts are presented below.

6.2.1 **VLSI Design.** The VLSI design results were the most disappointing of this research effort. Due to a combination of design errors, including a lack of design for testability, the results are inconclusive. No data contradicts the possibility that the circuit functions correctly when properly probed. However, there are three distinct possibilities why it was not possible to read a "1" from the LPROM memory cell array. First, the over bloated *metal2* lines may have prevented sufficient laser energy from reaching the underlying diffusion wells. Second, the close proximity of the laser link to the ROM transistor may have disrupted the electrical characteristics of the silicon and prevented the *n*-channel from properly forming beneath the gate. Finally, the links may not have successfully formed despite their appearance. All three of these conclusions beg for additional research.

6.2.2 **Automated Programming Station.** The automated programming station design effort rode a roller coaster of successes and failures. The end result; however, is a system which appears to function adequately.

6.2.2.1 **Hardware.** The Florod optics box functioned adequately once the optics were modified. Due to the power losses through the various optics, the higher powered 6 watt Spectra-Physics 164 laser actually performed better than the original

choice of a 4 watt model 2020-04 could have. The two stage shutter performed flawlessly. In addition to the shutter timing circuitry presented in this report, the Uni-Blitz SD-10 shutter controller has its own internal synchronization inputs. The laser, optics, and shutter system was able to deliver the desired power at the desired pulse durations to the test circuit.

Given sufficient lighting, the ITEX FG-100 image processing boards performed well. The only hardware disappointment was the stepper motor error. Its randomness makes it hard to characterize and correct.

6.2.2.2 **Software.** Two words can describe the automation software; effective and slow. Even without trying to compensate for the mechanical errors of the stepper motors, the pattern recognition software would have to be invoked 68 times. Within specifications established as a result of this research effort, the pattern recognition system performs flawlessly but requires 2.5 minutes to correctly identify a given template. Template generation is simple but time consuming and needs to be performed for each new chip run to maintain correct scaling. Stepper motor error correction using a single pass error characterization is marginal at best. Pattern recognition before each laser blast is more effective but can require a substantially greater amount of time to program an entire circuit.

6.3 Recommendations

Numerous recommendations are spawned by the results of this thesis. Prior to refabricating the entire LPROM, the established laser link parameters have to be verified using the actual laser programming hardware. This will require additional research and is covered in more detail in the following sections. Improvements to the existing LPROM design are presented first followed by the recommended improvements to the programming station and the automation software.

6.3.1 LPROM VLSI Design.

6.3.1.1 **Design for Testability.** The design errors, such as the shorts to ground and the missing connection to ground, need to be corrected. In addition, much greater testability needs to be incorporated into the circuit. Probe pads should be added to all the address and output data lines including the PRECHARGE and PRECHARGEbar lines. As space permits, probe pads should also be added to the *bitlines* and *wordlines*.

6.3.1.2 **Laser Programmable Column Sign Bit.** Another recommended improvement is to integrate a laser programmable *column sign bit into the sense amplifier*. This device, coupled with the appropriate software, has the potential to greatly reduce the total number of laser blasts necessary. Columns with more than half of their contents as "*1's*" would be inverted and the column sign bit programmed appropriately. Figure 6.1 below is a CIF plot of one proposed design. One of the two laser links must be programmed to connect the sense amplifier to the output bus.

6.3.1.3 **Dedicated Alignment Mark.** Another recommendation calls for generating and *installing a unique alignment mark* at the four corners of the LPROM array and at the beginning of each row. This would reduce the number of templates to one. In addition, a non-manhattan design would stand out against the rest of the array and be easily identified by the pattern recognition software.

6.3.2 Recommended Programming Station Hardware Improvements.

6.3.2.1 **X-Y Translation Stage.** A more reliable and accurate means of positioning the circuit beneath the laser needs to be developed. This could be ideally performed by an Anarad X-Y translation table. This system is capable of 0.2 μm positioning accuracy and provides very fast feedback on every movement. This table would obviate the use of pattern recognition at the beginning of each row while simultaneously ensuring accurate placement on every blast point. According to Lincoln Laboratories, theoretical programming speed could increase to 20 memory locations per second [Lin88].

If a new positioning system, such as the Anarad table, is not integrated into the laser programming station, the existing stepper motor system should examined and the

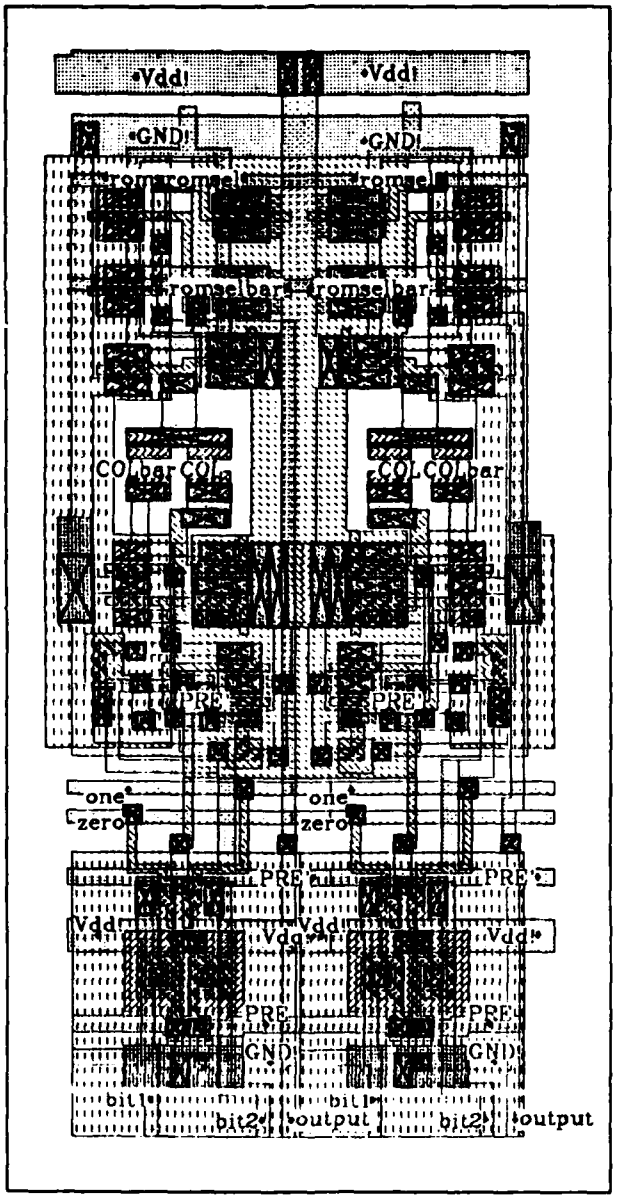


Figure 6.1. LPROM Sense Amplifier and Multiplexer with Column Sign Bit Inverters.

sources of stepping errors corrected. The controller, power supply and stepper motor are all suspect.

6.3.2.2 **Optics Box Autofocussing.** Due to the sensitivity of the optics to changes in separation between the objective and the circuit, autofocussing should be integrated into the optics box. The ITEX FG-100 image processing firmware could be used in conjunction with the host computer to drive a servo attached to the fine focus adjustment on the optics box.

6.3.2.3 **Beam Chopper.** Beam choppers are available that can integrate with the SD-10 shutter controller and provide infinite pulse duration settings. These units are precision balanced so that they can be placed on the floatation table along with the rest of the optics.

6.3.2.4 **Improved Laser Optics.** To reduce the loss of power, future laser programming station configurations should only use optics designed for the operating frequency of the laser. Due to the inevitable loss of power through the optics, a 5 watt laser such as a Spectra-Physics 2020-05 or 2016-05 should be used. This will ensure the minimum 3.5 watts out the objective.

6.3.3 Recommended Software Improvements.

6.3.3.1 **Array Processor.** To reduce the total time spent performing pattern recognition, a faster computer system equipped with an array processor, such as a SUN4 should be used. Preliminary estimations reduce the correlation times to under 2 seconds.

6.3.3.2 **Column Sign Bit Software.** Additional software needs to be integrate into the automation system to support the laser programmable column sign bits. To illustrate the change to output data, Table 6.1 below illustrates the data inputted for formatting, the program array without column *sign bits* in the form of *out_array* and, the data modified with column *sign bits* in the array *final_out*. Also included in this table is a

row of bits which represent the *sign bits* to be programmed in the sense amplifiers. A *"1"* in the row of *sign bits* represents a sense amplifier with inverted outputs.

input array

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31

out_array()

00000111111001101000000000100110000010000
00110111110011011100001101001100000011100
11000111110011010011110001001100000010011
11110111110011011111111101001100000011111
00011111000000110100000100110000000000100
11011111000000110111110100110000000000111
00011111111100110100000100111111100000100
11011100111111000111110100100111010100 11

sign_array()

00001111110011110000000000001100000000000

final_out()

00001000000000100000000001000000000010000
00111000000000101100001101000000000011100
11001000000000100011110001000000000010011
11111000000000101111111101000000000011111
00010000110011000100000100111100000000100
11010000110011000111110100111100000000111
00010000001111000100000100110011000000100
11010011001100110111110100101011010100 11

Table 6.1. Column Sign Bit Data.

6.4 Recommended Follow-On Research Efforts

6.4.1 **AFIT Laser Link Parameterization.** Because of the numerous deviations from the original design. A parameterization of the AFIT laser link needs to be performed. This includes generating a series of test circuits with varying widths and gaps and experimenting with different pulse durations, laser powers and possibly different types of lasers. In addition, MOS transistors should be placed at varying proximities to these links to determine what effects, if any, local heating has on their operation.

6.4.2 **Laser Link Reliability.** A study should be performed to determine the long range reliability of laser diffused and laser cut circuits. The effects of blasting away the protective overglass to circuit lifetime will help determine the applications of this programming technique.

6.4.3 **Additional Applications of the Laser Programming Station.** The ability to precisely place a laser pulse of varying strength, duration, and size should find applications in other areas of VLSI research. It would be beneficial to determine additional uses for this piece of test equipment.

Appendix A. *SPICE Simulation Data*

A.1 Introduction

This appendix provides the SPICE decks and technology files for the schematics provided in Chapter 3.

```
SIMULATION OF THE LPROM SENSEAMP and CELL--1.2UM CMOS, WFT16 LOADING

*CHAIN OF 17 lamda wide trans

.SUBCKT CHAIN2 1 2 3
* 1=VDD 2=INPUT 3=OUTPUT
MN1 4 1 2 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MN2 5 1 4 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MN3 6 1 5 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MN4 7 1 6 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MN5 8 1 7 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MN6 9 1 8 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MN7 3 1 9 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
C4  4  0  .01P
C5  5  0  .01P
C6  6  0  .01P
C7  7  0  .01P
C8  8  0  .01P
C9  9  0  .01P
.ENDS CHAIN2

*INPUT INVERTER TO PLA COLUMN

*MN1 3 2 0 0 HP_NM1_DU1 W=28.8U L=1.2U AD=13P
*MP1 3 2 1 1 HP_PM1_DU2 W=38.4U L=1.2U AD=47.5P
MN1 3 2 0 0 HP_NM1_DU1 W=86.8U L=1.2U AD=26P
MP1 3 2 1 1 HP_PM1_DU2 W=126.4U L=1.2U AD=96.5P

*ACTIVE PROD TERM PULLDOWN & PULLUP

MN2 4 3 0 0 HP_NM1_DU1 W=10.2U L=1.2U AD=14P AS=14P PD=7.2U PS=7.2U
MP3 6 0 1 1 HP_PM1_DU2 W=5.4U L=3.0U
*MP03 6 15 1 1 HP_PM1_DU2 W=3.6U L=1.2U AD=5P

*LOAD INVERTER ON PLA COLUMN

MN3 0 3 0 0 HP_NM1_DU1 W=375U L=1.2U
```

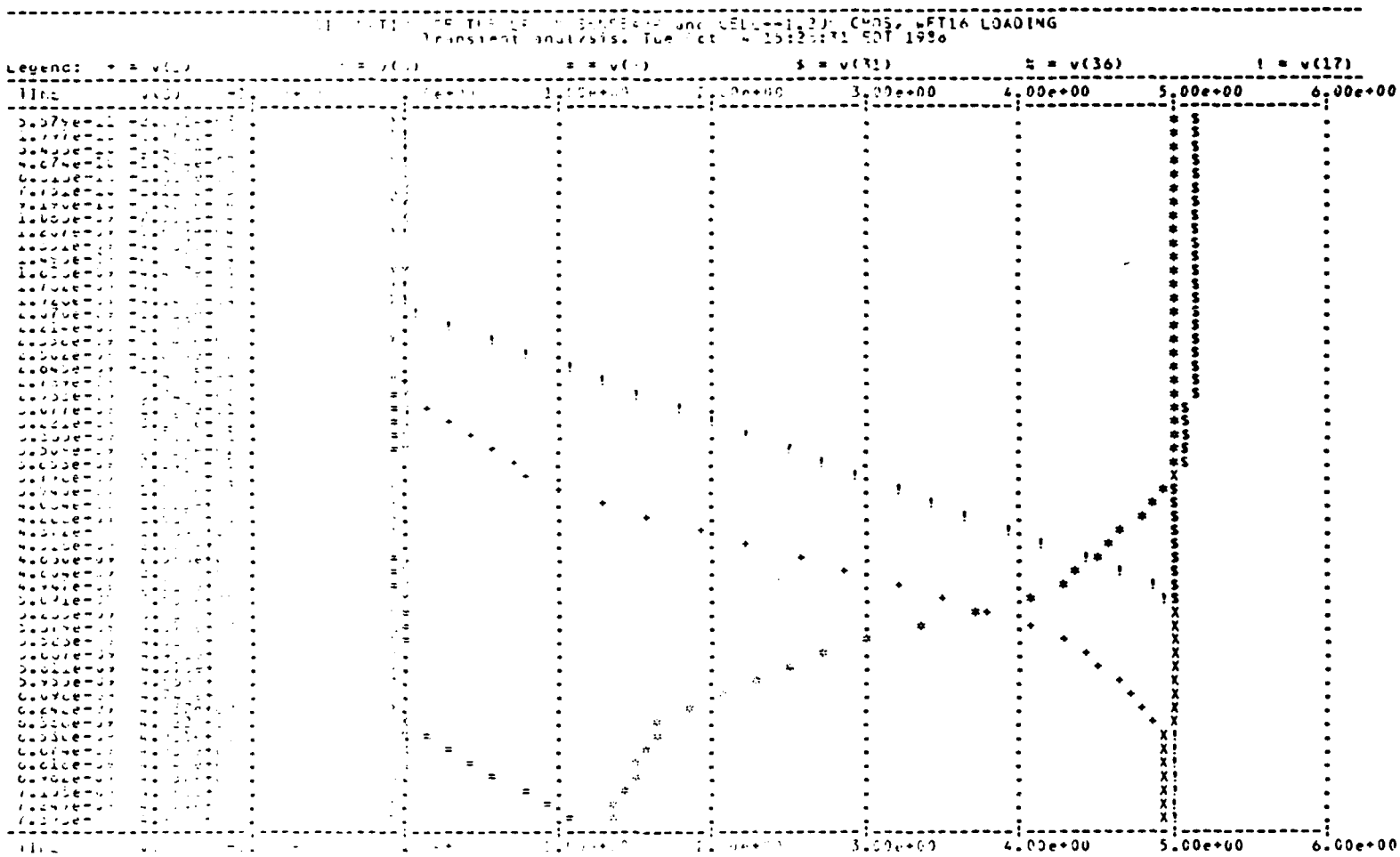
```
*OUTPUT INVERTERS DRIVING WORD LINE

MN54 55 6 0 0 HP_NM1_DU1 W=1.8U L=1.2U AD=9P    PD=9U
MP54 55 6 1 1 HP_PM1_DU2 W=6.0U L=1.2U AD=19.5P PD=12.6U

MN4 56 55 0 0 HP_NM1_DU1 W=5.4U L=1.2U AD=16.2P PD=11.4U
MP4 56 55 1 1 HP_PM1_DU2 W=9.6U L=1.2U AD=28.8P PD=15.6U

MN53 7 56 0 0 HP_NM1_DU1 W=9.6U L=1.2U AD=28.8P PD=15.6U
MP53 7 56 1 1 HP_PM1_DU2 W=19.2U L=1.2U AD=57.6P PD=25.2U

* XROMCELL GATE FROM DRIVEN WORDLINE

MN30 70 8 31 0 HP_NM1_DU1 W=1.8U L=1.2U AD=7.2P PD=7.8U

*BITLINE PRECHARGE

MP31 31 16 1 1 HP_PM1_DU2 W=12.U L=1.2U AD=9.0P

*SIMULATED 4:1 MUX TRANSISTOR

MN31 31 1 32 0 HP_NM1_DU1 W=8.4U L=1.2U AD=7.2P

* PRE driver/inverter to sim PRE'

MP40 16 17 1  1 HP_PM1_DU2 W=150U L=1.2U AD=30.0P PD=12U
MN40 16 17 0  0 HP_NM1_DU1 W=75U L=1.2U AD=15.0P PD=9.0U

* SIMULATED LOAD on PRE'

MP41 0 16 0  1 HP_PM1_DU2 W=259.8U L=3.0U AD=432.0P PD=172.8U

* SENSE AMP SIMULATION

MP32 32 16 1 1 HP_PM1_DU2 W=6U L=1.2U AD=14.4P PD=15U
MP33 32  0 1 1 HP_PM1_DU2 W=1.8U L=3.0U AD=3P PD=1.2U

MP34 33 32 1  1 HP_PM1_DU2 W=7.2U L=1.2U AD=30.0P PD=12U
MN32 33  1 0  0 HP_NM1_DU1 W=1.8U L=4.8U AD=6.84P PD=9.0U

*OUTPUT INVERTERS  AND WORD SIGN MUX

MP35 34 33 1  1 HP_PM1_DU2 W=9.6U L=1.2U AD=19.8P PD=16.0U
MN35 34 33 0  0 HP_NM1_DU1 W=4.8U L=1.2U AD=14.4P PD=18.0U
```

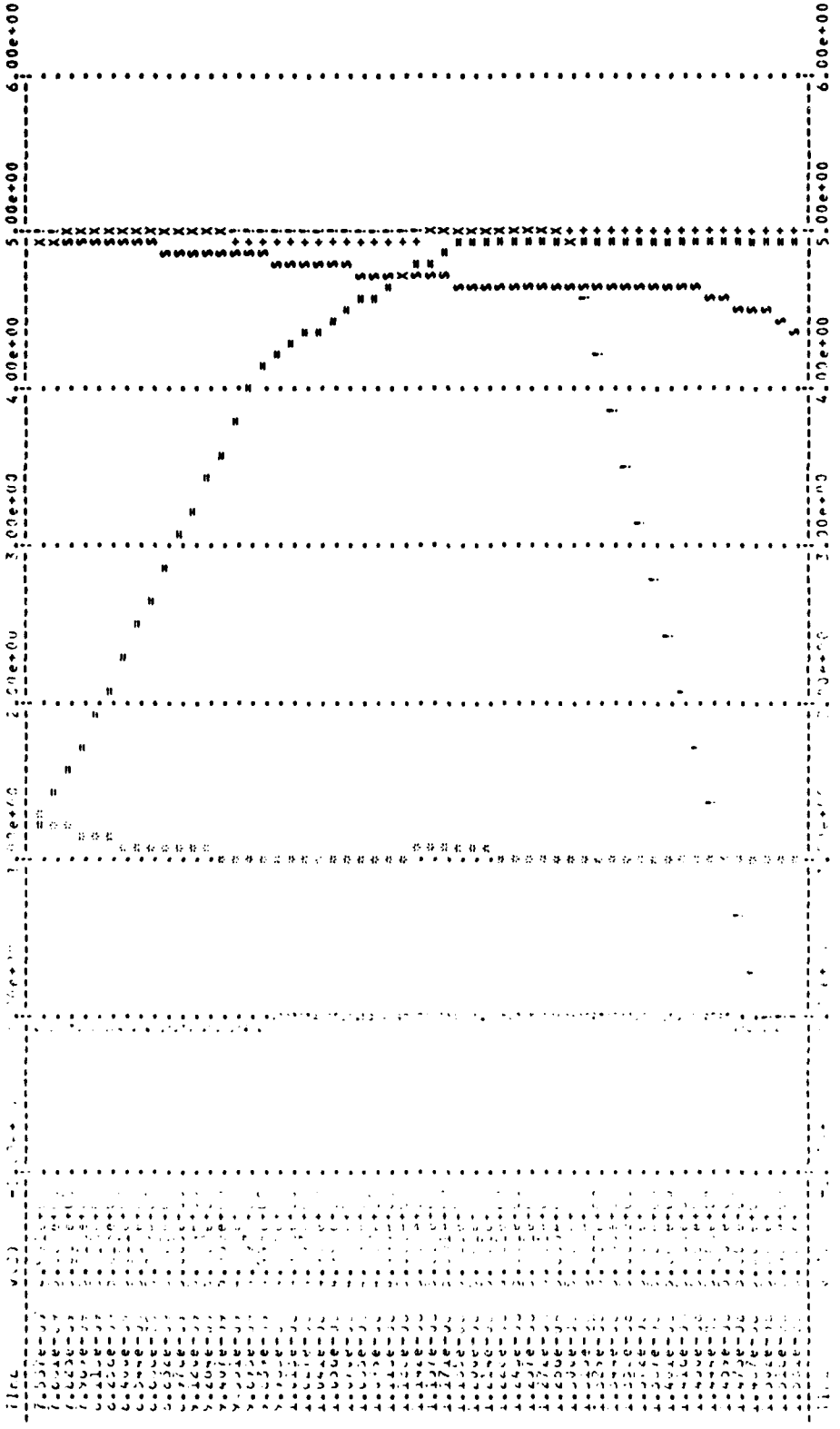
```
MP36 35 34 1  1 HP_PM1_DU2 W=9.6U L=1.2U AD=19.8P PD=16.0U
MN36 35 34 0  0 HP_NM1_DU1 W=4.8U L=1.2U AD=14.4P PD=18.0U

MN37 36 38 35 0 HP_NM1_DU1 W=3.6U L=1.2U AD=8.64P PD=3.6U
MP37 36 37 35 1 HP_PM1_DU2 W=7.2U L=1.2U AD=12.96P PD=3.6U

MN38 36 37 35 0 HP_NM1_DU1 W=3.6U L=1.2U AD=8.64P PD=3.6U
MP38 36 38 35 1 HP_PM1_DU2 W=7.2U L=1.2U AD=12.96P PD=3.6U

* OUTPUT LOAD

MN62 0 34 0 0 HP_NM1_DU1 W=3.0U L=1.2U

* INTERCONNECT PARASITICS

MN42 0 31 0 0 HP_NM1_DU1 W=259U L=1.2U
MN21 0 8 0 0 HP_NM1_DU1 W=317U L=1.2U
CWORD 8 0 .06P
RLPROM 70 0 1000
RWORD 7 8  60
CWORD7 7 0 .05P
C34 34 0 .05P
C36 36 0 .05P
CPLACOL 3 0 .2P
CCHAIN 4 0 .05P
C55 55 0 .01P
C56 56 0 .01P
C6 6 0 .01P
*INSERT CHAIN
X1 1 4 6 CHAIN2

*INPUT VOLTAGE SOURCES
VWS0 37 0  DC 0
VWS1 38 1  DC 5
VDD 1 0 DC 5
VIN 2 0 PULSE(5 0 2N 3N 3N 37N 80N)
VPRE 17 0 PULSE(0 5 2N 3N 2N 8N 40N)

.TRAN 1N 100N
.WIDTH OUT=80
*** OPTIONS ***
.model HP_du2 r rsh = 130 defw = 0 narrow = 0
.model HP_du2 c cj = .00053 cjsw = 2.8e-10 defw = 0 narrow = 0
.model HP_du1 r rsh = 80 defw = 0 narrow = 0
```

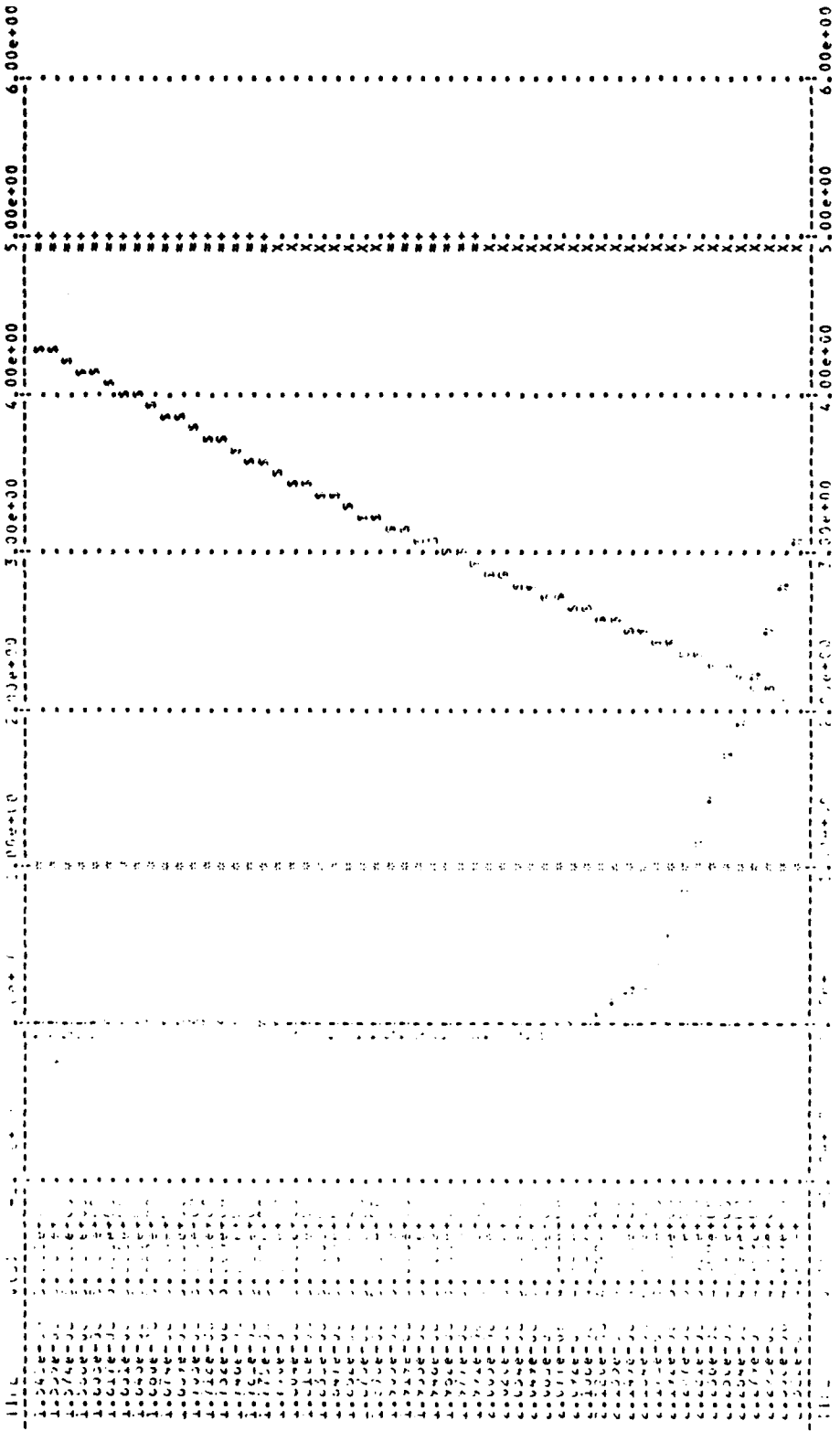
```
.model HP_du1 c cj = .00032 cjsw = 2.7e-10 defw = 0 narrow = 0
.model HP_pm1_du2 pmos level=4
+ vfb = -.22542 lvfb = -.0019813 wvfb = .0173064
+ phi = .74558 lphi = 0 wphi = 0
+ k1 = .458908 lk1 = -.046638 wk1 = .0701431
+ k2 = -.025004 lk2 = .0228713 wk2 = -.0024942
+ eta = -.0071162 leta = .018817 weta = .003537
+ muz = 157.966 dl = .22247 dw = 1.23813
+ u0 = .144794 lu0 = .0434678 wu0 = -.040423
+ u1 = -.0077448 lu1 = .106373 wu1 = .00204049
+ x2mz = 7.74324 lx2mz = -1.6053 wx2mz = 2.04544
+ x2e = -.001029 lx2e = -.00064983 wx2e = -.0012181
+ x3e = .000614143 lx3e = -.0008 wx3e = -.0013628
+ x2u0 = .00826952 lx2u0 = -.00061654 wx2u0 = .00104945
+ x2u1 = -.00043784 lx2u1 = .00752637 wx2u1 = -1.4648e-05
+ mus = 154.336 lmus = 47.4389 wmus = 5.38354
+ x2ms = 6.8254 lx2ms = 3.74492 wx2ms = 2.39692
+ x3ms = .559784 lx3ms = 1.35373 wx3ms = -.42505
+ x3u1 = .00115368 lx3u1 = -.011678 wx3u1 = -.00066211
+ tox = .0163 temp = 27 vdd = 5
+ cgdo = 2.33e-10 cgso = 2.33e-10 cgbo = 2.595e-09
+ xpart = 1
+ n0 = 1 ln0 = 0 wn0 = 0
+ nb = 0 lnb = 0 wnb = 0
+ nd = 0 lnd = 0 wnd = 0
+ rsh = 130 cj = .00053 cjsw = 2.8e-10
+ js = 1.e-08 pb = .9 pbsw = .9
+ mj = .5 mjsw = .33 wdf = 0
+ dell = 0
.model HP_pm1_du1 pmos level=4
+ vfb = -.22542 lvfb = -.0019813 wvfb = .0173064
+ phi = .74558 lphi = 0 wphi = 0
+ k1 = .458908 lk1 = -.046638 wk1 = .0701431
+ k2 = -.025004 lk2 = .0228713 wk2 = -.0024942
+ eta = -.0071162 leta = .018817 weta = .003537
+ muz = 157.966 dl = .22247 dw = 1.23813
+ u0 = .144794 lu0 = .0434678 wu0 = -.040423
+ u1 = -.0077448 lu1 = .106373 wu1 = .00204049
+ x2mz = 7.74324 lx2mz = -1.6053 wx2mz = 2.04544
+ x2e = -.001029 lx2e = -.00064983 wx2e = -.0012181
+ x3e = .000614143 lx3e = -.0008 wx3e = -.0013628
+ x2u0 = .00826952 lx2u0 = -.00061654 wx2u0 = .00104945
+ x2u1 = -.00043784 lx2u1 = .00752637 wx2u1 = -1.4648e-05
+ mus = 154.336 lmus = 47.4389 wmus = 5.38354
+ x2ms = 6.8254 lx2ms = 3.74492 wx2ms = 2.39692
```

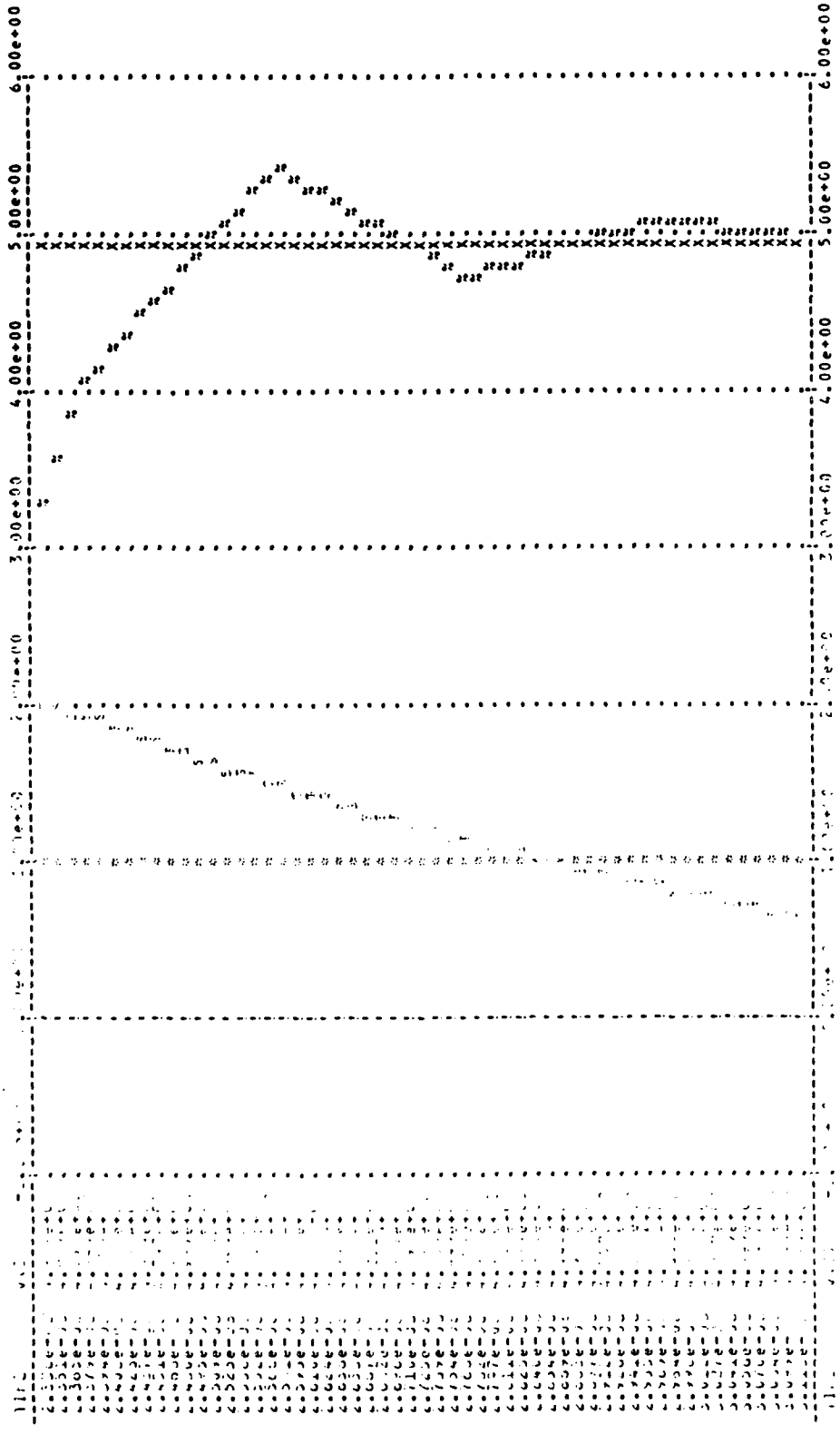
```
+ x3ms = .559784 lx3ms = 1.35373 wx3ms = -.42505
+ x3u1 = .00115368 lx3u1 = -.011678 wx3u1 = -.00066211
+ tox = .0163 temp = 27 vdd = 5
+ cgdo = 2.33e-10 cgso = 2.33e-10 cgbo = 2.595e-09
+ xpart = 1
+ n0 = 1 ln0 = 0 wn0 = 0
+ nb = 0 lnb = 0 wnb = 0
+ nd = 0 lnd = 0 wnd = 0
+ rsh = 80 cj = .00032 cjsw = 2.7e-10
+ js = 1.e-08 pb = .8 pbsw = .8
+ mj = .95 mjsw = .12 wdf = 0
+ dell = 0
.model HP_nm1_du2 nmos level=4
+ vfb = -1.0298 lvfb = -.0093555 wvfb = .0647494
+ phi = .843816 lphi = 0 wphi = 0
+ k1 = 1.12483 lk1 = -.0021092 wk1 = -.049443
+ k2 = .135923 lk2 = .0302784 wk2 = -.044754
+ eta = -.0039367 leta = .00848664 weta = .00332115
+ muz = 467.77 dl = .387011 dw = 1.17123
+ u0 = .0684497 lu0 = .0883554 wu0 = -.035214
+ u1 = .0195292 lu1 = .161764 wu1 = -.022371
+ x2mz = 7.13301 lx2mz = -3.4283 wx2mz = 11.9897
+ x2e = -.0018255 lx2e = -.0025663 wx2e = -.0015149
+ x3e = .00016001 lx3e = -.00013223 wx3e = -.0018577
+ x2u0 = .000314041 lx2u0 = .00211097 wx2u0 = .0038767
+ x2u1 = .00426367 lx2u1 = .00659641 wx2u1 = -.003162
+ mus = 519.946 lmus = 100.137 wmus = -50.002
+ x2ms = -2.1684 lx2ms = 10.7257 wx2ms = 16.0132
+ x3ms = 1.46977 lx3ms = 15.1963 wx3ms = -3.6649
+ x3u1 = -.0059905 lx3u1 = .0141446 wx3u1 = .00244578
+ tox = .0163 temp = 27 vdd = 5
+ cgdo = 4.093e-10 cgso = 4.093e-10 cgbo = 2.471e-09
+ xpart = 1
+ n0 = 1 ln0 = 0 wn0 = 0
+ nb = 0 lnb = 0 wnb = 0
+ nd = 0 lnd = 0 wnd = 0
+ rsh = 130 cj = .00053 cjsw = 2.8e-10
+ js = 1.e-08 pb = .9 pbsw = .9
+ mj = .5 mjsw = .33 wdf = 0
+ dell = 0
.model HP_nm1_du1 nmos level=4
+ vfb = -1.0298 lvfb = -.0093555 wvfb = .0647494
+ phi = .843816 lphi = 0 wphi = 0
+ k1 = 1.12483 lk1 = -.0021092 wk1 = -.049443
+ k2 = .135923 lk2 = .0302784 wk2 = -.044754
```

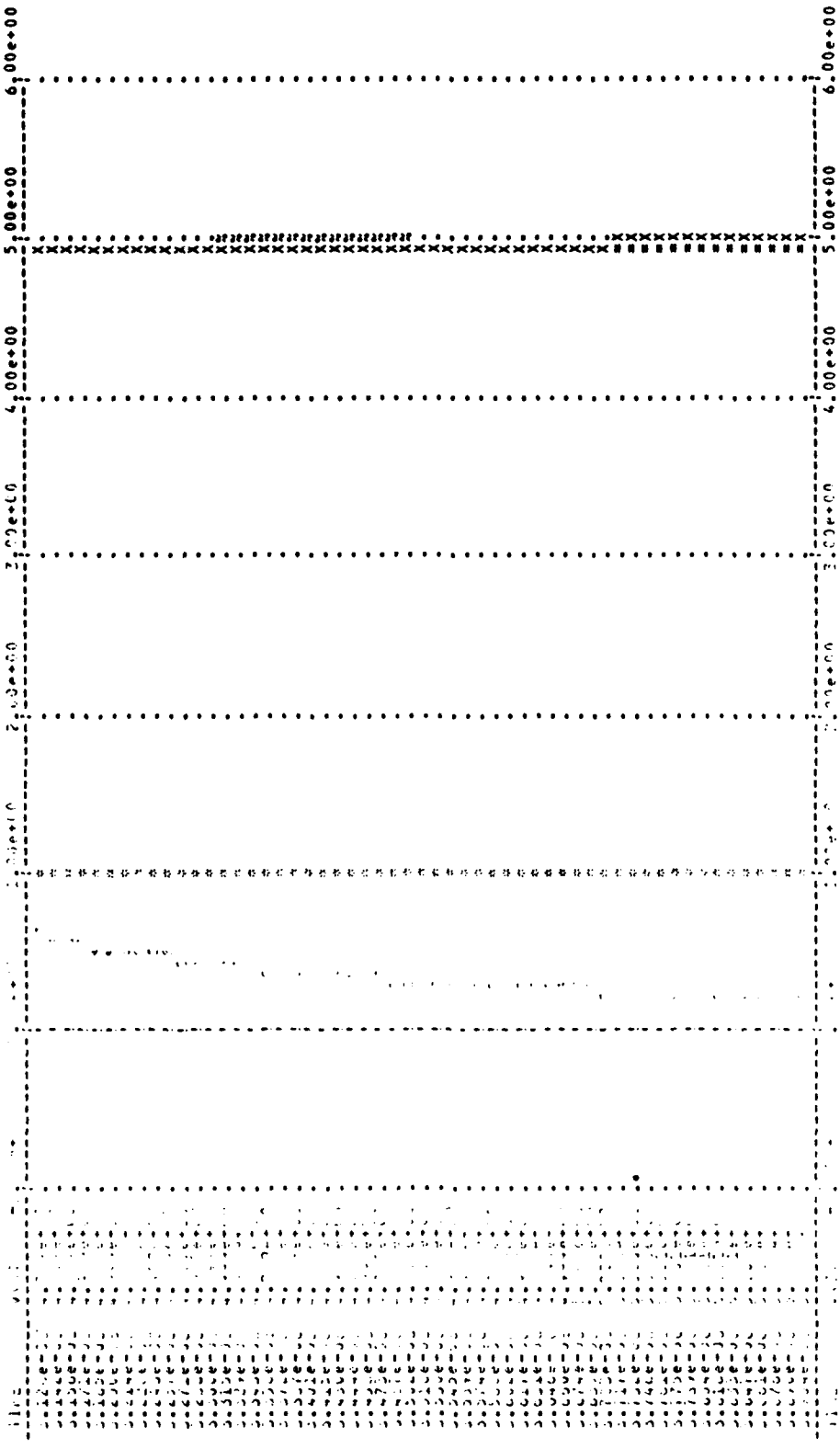
```
+ eta = -.0039367 leta = .00848664 weta = .00332115
+ muz = 467.77 dl = .387011 dw = 1.17123
+ u0 = .0684497 lu0 = .0883554 wu0 = -.035214
+ u1 = .0195292 lu1 = .161764 wu1 = -.022371
+ x2mz = 7.13301 lx2mz = -3.4283 wx2mz = 11.9897
+ x2e = -.0018255 lx2e = -.0025663 wx2e = -.0015149
+ x3e = .00016001 lx3e = -.00013223 wx3e = -.0018577
+ x2u0 = .000314041 lx2u0 = .00211097 wx2u0 = .0038767
+ x2u1 = .00426367 lx2u1 = .00659641 wx2u1 = -.003162
+ mus = 519.946 lmus = 100.137 wmus = -50.002
+ x2ms = -2.1684 lx2ms = 10.7257 wx2ms = 16.0132
+ x3ms = 1.46977 lx3ms = 15.1963 wx3ms = -3.6649
+ x3u1 = -.0059905 lx3u1 = .0141446 wx3u1 = .00244578
+ tox = .0163 temp = 27 vdd = 5
+ cgdo = 4.093e-10 cgso = 4.093e-10 cgbo = 2.471e-09
+ xpart = 1
+ n0 = 1 ln0 = 0 wn0 = 0
+ nb = 0 lnb = 0 wnb = 0
+ nd = 0 lnd = 0 wnd = 0
+ rsh = 80 cj = .00032 cjsw = 2.7e-10
+ js = 1.e-08 pb = .8 pbsw = .8
+ mj = .95 mjsw = .12 wdf = 0
+ dell = 0
*
.OPTIONS DEFL=1.2U DEFW=6U DEFAS=45P DEFAD=45P
+ITL1=500 ITL4=30 ABSTOL=100P VNTOL=100U CHGTOL=1E-12
+NOPAGE RELTOL=.004 CPTIME=15000
.END
```

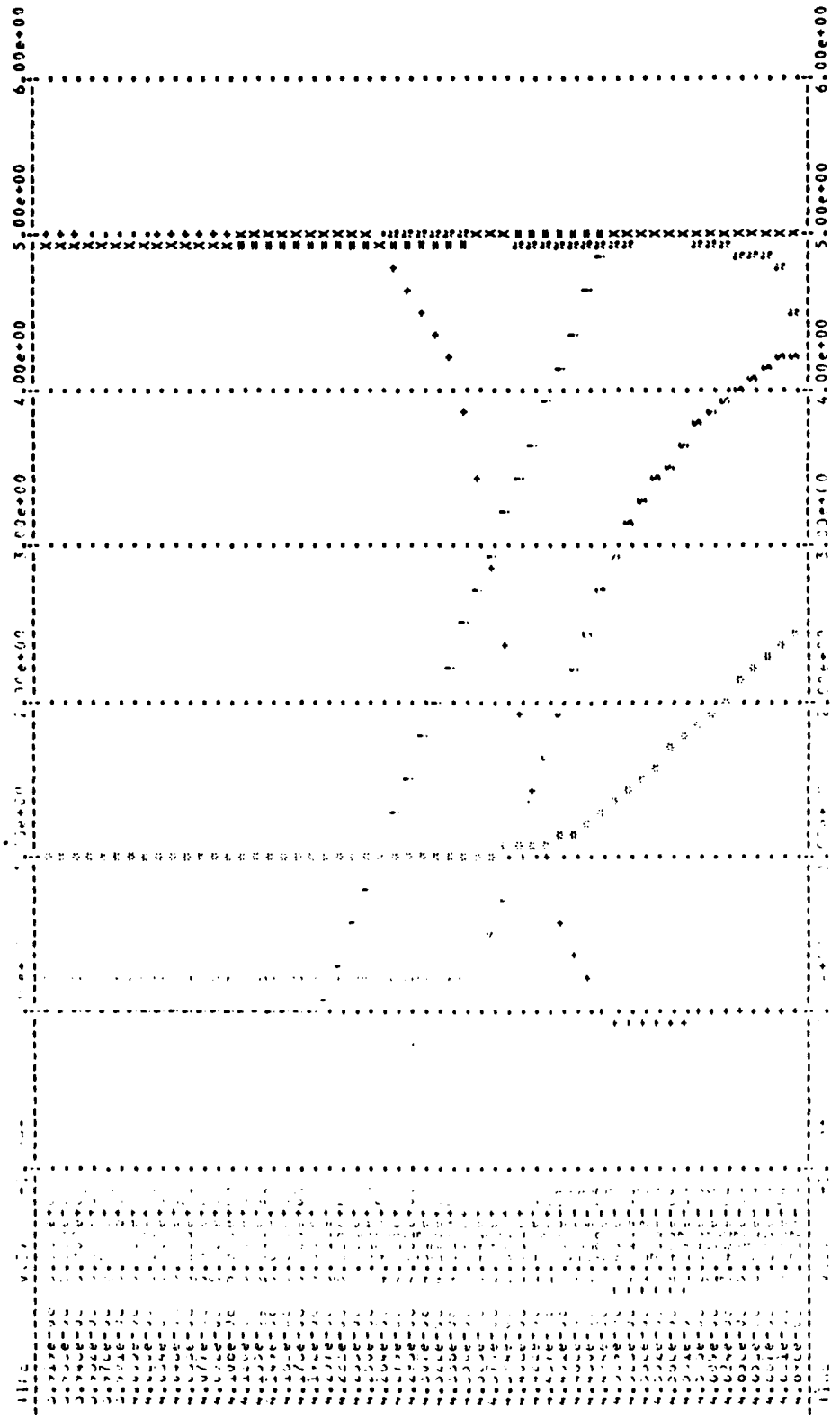

Legend: + = v(2) = v(3) = v(4) $ = v(31) % = v(36) ! = v(17)
Time
3.00e+00
4.00e+00
5.00e+00
6.00e+00

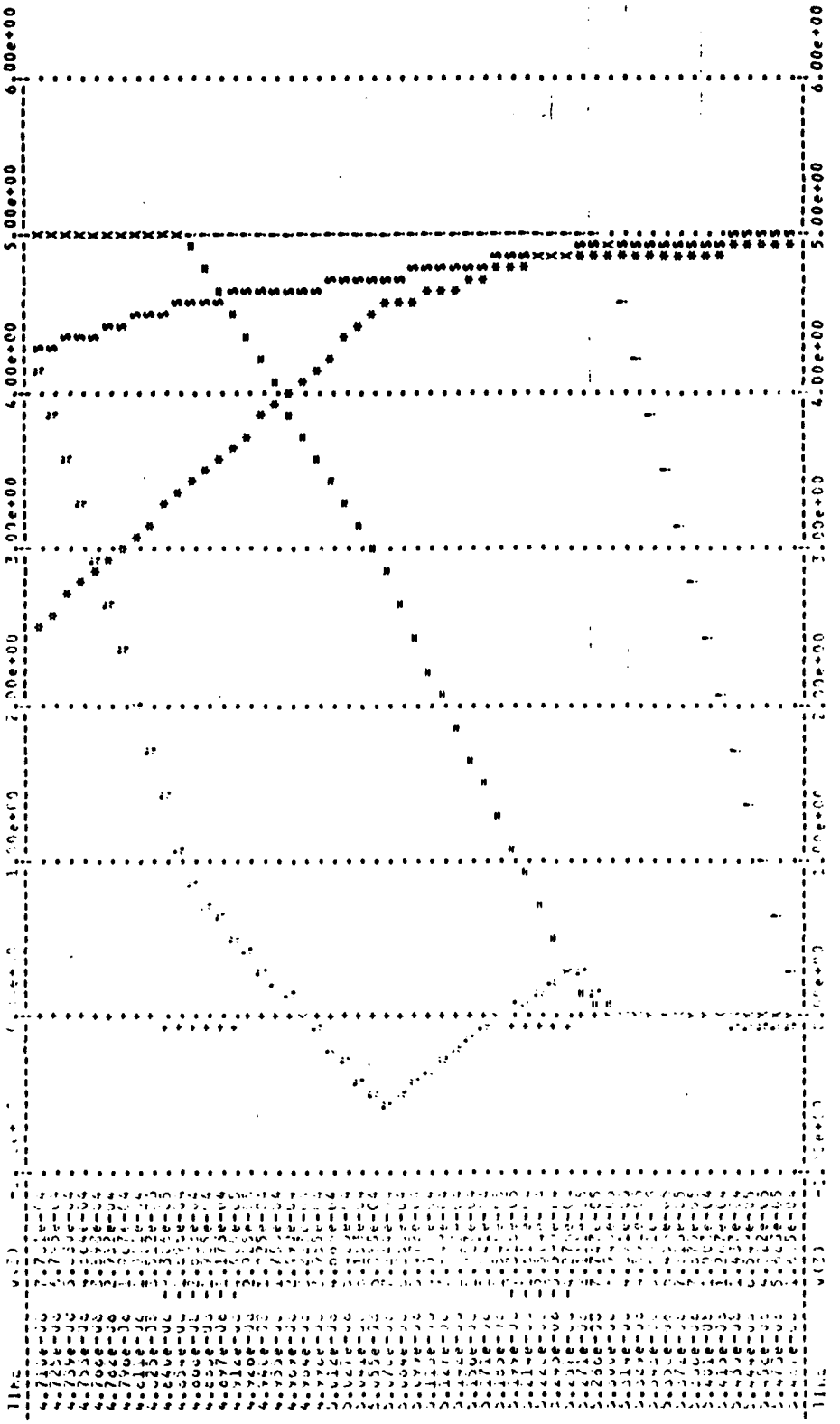

6.00e+00
5.00e+00
4.00e+00
3.00e+00


6.00e+00
5.00e+00
4.00e+00
3.00e+00


6.00e+00
5.00e+00
4.00e+00
3.00e+00


6.00e+00
5.00e+00
4.00e+00


6.00e+00
5.00e+00
4.00e+00
6.00e+00
5.00e+00
4.00e+00



Appendix B. *Stepper Motor Operation*

B.1 Introduction

This appendix is provided as a supplement to the Superior Electric Modulynx Controller *Operations manual which is somewhat difficult to decipher.* This appendix covers those features of the Modulynx controller which are pertinent to the operation of the automated laser programming station.

B.2 Modulynx Controller Power-Up

The following is the sequence of events to be followed during normal power-up of the Modulynx Mitas stepper motor controller. These steps must also be followed when resetting the device (there is no reset switch, reset is accomplished by cycling the power switch). Pausing or terminating the execution of a program on the host computer as it executes a subroutine call to the Modulynx controller will freeze the execution command. *The controller must be reset to continue.*

- Upon power-up the Modulynx stepper motor controller will prompt for "PROGRAM CONTROLS". Depress the number 5 to initiate PROGRAM CONTROLS..
- A configuration number "3010" will appear. This default setting is correct. Depress the right arrow 4 times to cycle through the digits and proceed to the next command.
- The controller should now prompt "CLR". Depress the number "0".
- The controller now prompts "LRN" Depress the number "0".
- The controller now prompts for RS-232 mode. Depress the number "2" to ready the controller to receive commands from the MicroVaxII computer.
- The prompt "ARE YOU SURE?" should now be seen. Depress the number "4" to confirm RS-232 mode.
- The controller now prompts for BAUD RATE. Depress the number "0" to proceed to the next selection of baud rates.
- Select the 600 baud rate by depressing the number "6".

The controller should now display "RS-232 MODE" which signifies it is ready to receive commands from the host computer.

B.3 Modulynx Controller Settings

The following settings were experimentally determined to be optimum for use in the Automated LPROM Laser Programming Station. Values of R/U greater than 0.005 caused the stepper motors to malfunction and not step at all.

PARAMETER	SETTING	COMMENT
XM_CODE	111	Type X of Motor.
X R/U	.005	Internal Scale Factor.
X_Z	000100.00	Offset From Motor Stop.
YM_CODE	111	Type Y of Motor.
Y R/U	.005	Internal Scale Factor.
Y_Z	000100.00	Offset From Motor Stop.
X BASE SPEED	12.5	Lowest Motor Speed.
X MAX ACCEL	5000	Accel from BASE to HI.
X MAX DECEL	5000	Decel from HI to BASE.
X BACK	0000	X-dir Backlash Setting.
Y BASE SPEED	12.5	Lowest Motor Speed.
Y MAX ACCEL	5000	Accel from BASE to HI.
Y MAX DECEL	5000	Decel from HI to BASE.
Y BACK	0000	Y-dir Backlash Setting.

Table B.1. Modulynx Controller Settings.

B.4 Changing the Modulynx Controller Settings

One of the nice features of the Modulynx controller is its internal calculations of it motor limitations. For a given internal scale factor, R/U, the controller will calculate a *maximum* BASE, HI, ACCEL, and DECEL.

Entering a very large value for a given parameter will cause the controller to reject this value and indicate the largest value it is able to accept.

To change the parameters depress "0" when prompted for RS-232 mode. This will enter you into the parameters mode. Use the right and left arrows to transition through

numbers you don't want changed. When all the parameters are inputted, the left arrow will step you back to the RS-232 prompt. Continue as indicated above to ready the controller for receipt of instructions from the host computer.

B.5 Stepper Motor Positioning

Once the controller and motor power supplies are powered up, the motors will freeze in their present location. Consequently, the circuit has to be aligned and positioned prior to power-up. If repositioning is required, turn off the motor power supply, reposition the circuit and repower the stepper motor. You can force the shaft to rotate, but this is highly discouraged as it generates excessive wear on the internal gears and bearings.

Appendix C. *ITEX FG-100 Hardware and TEST100 Software Operation*

C.1 Introduction

This appendix provides a detailed summary of the steps necessary to generate the pattern recognition templates used in the automated laser programming station, including the video hardware configuration. The ITEX FG-100 programmers manual is an excellent reference for software questions not addressed in this appendix.

C.2 Video Hardware Configuration

C.2.1 **ITEX FG-100 Boards.** The center of the video hardware is the ITEX FG- 100 image processing boards. These boards are mounted in the MicroVaxII, SMV2A, located in building 640, room 241. The inputs to these boards are wired into the patch panel mounted in the 2 rolling 19 inch equipment racks. The video outputs from these boards are also wired into this patch panel.

C.2.2 **Video Cameras.** The other major piece of hardware is the video camera. The Florod optics box came complete with a RCA close circuit black and white camera. The camera uses a "C" mount and includes a C-mount to optics box adapter. If this camera is used, the coaxial cable output is patched into the ITEX FG-100 video inputs directly. The alternative camera is the JVC RC110. This camera normally accepts a bayonet style lens, but comes equipped with a C-mount adapter to facilitate mounting to the optics box. If this camera is used, the output is patched via the 14 pin RS-110 cable to the back of the JVC RS-110 remote control unit. This unit is mounted adjacent to the patch panel. Its outputs are also wired to this panel.

C.2.3 **Monitors.** The inputs for both the black and white and color monitors are wired to the patch panel. Because there is no color visible from the circuits through either of the cameras, the black and white monitor delivers the sharper of the two images.

C.2.4 **Camera Settings.** There are four filters on a rotating thumb wheel built into the JVC camera. Filter #2 works the best for view the VLSI circuits. The remote

control unit should be switched to the 12 db setting. If the image is too bright, the camera will attempt to reduce the gain and cycle through gain limits. If the camera fails to stop cycling, reduce the intensity of the illuminating bulb by reducing the voltage output from the power supplies. There are no adjustments on the RCA camera.

C.2.5 **Optics Box Video Configuration.** There are four items which must be addressed when configuring the video portion of the Florod optics box. The type and placement of the optical filters, and the positions of the binocular head, C-mount adapter, and light bulb must be accounted for before pattern recognition or laser programming is accomplished. Figure C.1 below illustrates the position of these various parts. The aligning of the different objectives is covered in Appendix D.

C.2.5.1 **Type and Position of Optical Filters.** There are two primary filter configurations. The first places the orange safety filter at the top of the beam splitter/prism assembly located at the base of the binocular head. This configuration protects both the observer and camera for laser light reflected back up through the objective. The second configuration uses a different beam splitter/prism assembly with the safety filter removed. This configuration should only be used for the low power laser microscope. Additional safety lenses should be placed in the eye pieces to protect the observer.

Replacing these assemblies requires the removal of the top plate of the optics box and unscrewing the prism from the front plate. Be careful not to damage the dielectric mirror. Realignment of the laser may be required.

C.2.5.2 **Illumination Bulb Position.** The position of the highest intensity light through the objective onto the circuit is a function of the position of the illumination bulb behind the beam splitter/prism assembly. The bulb holder resides in a black spacer ring which floats with one degree on freedom in a mount screwed to the top plate of the optics box. Positioning is best accomplished by accessing the bulb through the top, side access panel. The one provided adjustment can be made with an allen wrench through a set screw located on the lower corner of the mount. Additional adjustments can be

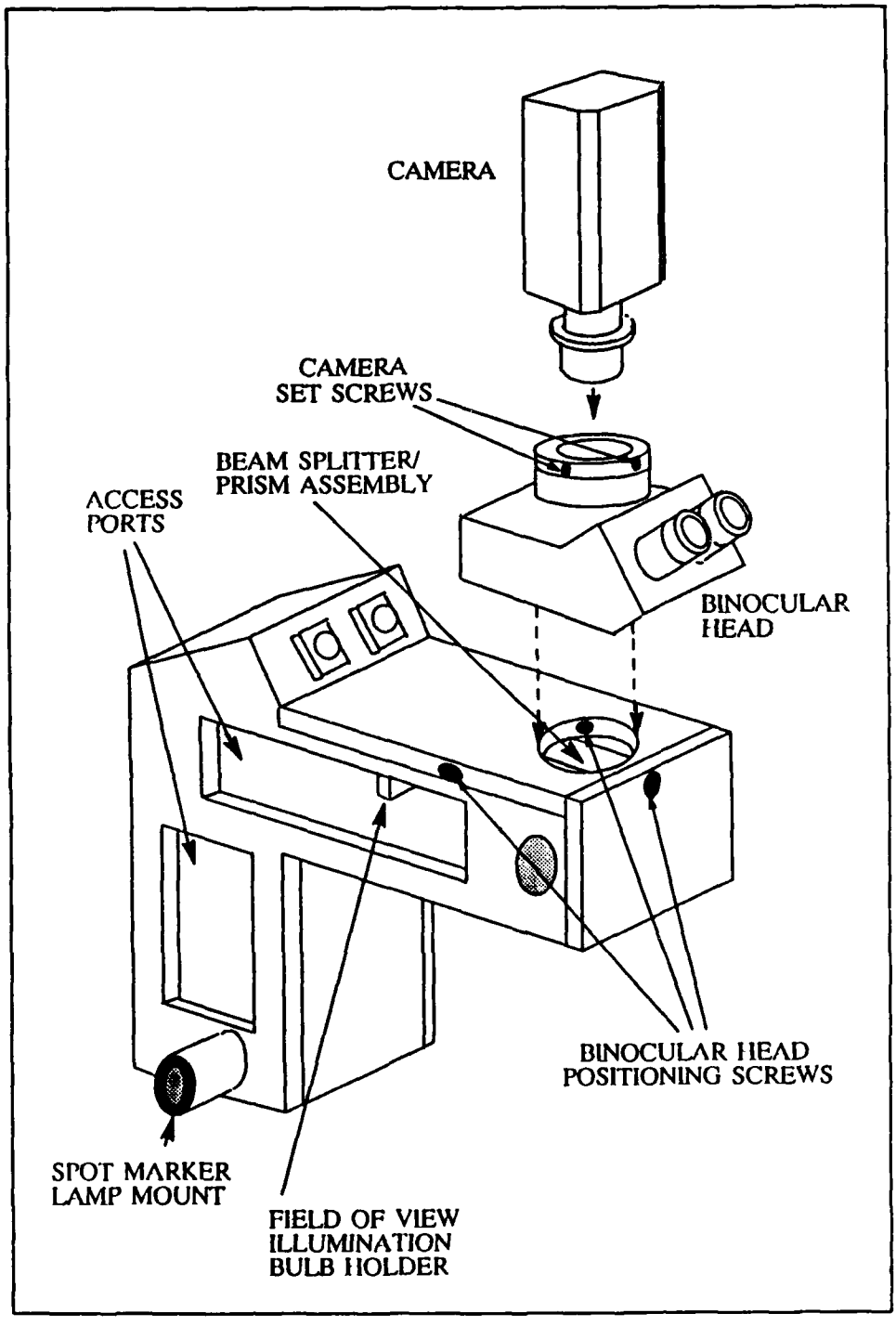


Figure C.1. Florod Optics Box Video Configuration.

made by carefully rotating and angling the back of the bulb holder. Proper illumination is critical to the operation of the ITEX FG-100 boards.

C.2.5.3 **Position of Binocular Head.** The binocular head should be positioned to center the view of a laser zap on the monitor. The position of the laser beam should never be adjusted to meet the field of view of the camera. Three set screws in a "Y" configuration control the position of the binocular head. These too require an allen wrench. A cross hair drawn on the monitor is useful in centering the field of view.

C.2.5.4 **Mounting Camera on Optics Box.** Either camera will have to use the C-mount to optics box adapter. Once the adapter is securely attached to the camera, the adapter must be inserted into the top of the binocular head. Once the scene from the camera is correctly aligned with the X-Y movement of the translation stage, its position should be secured by tightening the two set screws located on the top black ring of the binocular head.

C.3 Generating Pattern Recognition Templates.

Generating pattern recognition templates is an exercise in using the TEST100 software. All the required commands are found in the numerous subroutine menus.

After loggin in, this software is accessed by typing "TEST100 <CR>".

The first prompt will be the hardware configuration and initialization. The default values should be correct. See the ITEX programmers guide if these parameters need to be modified. Entering the number "0" will take you to the Main Menu. Entering "0" will always take you from one menu to the next higher menu in the menu hierarchy. Entering "0" from the Main Menu will exit you from TEST100.

If the image is not visible on the screen, check the wiring of the patch panel to ensure the output of the ITEX boards is being fed into the monitor's inputs. Next enter the number "4" from the Main Menu. This will take you to the image capturing menu. Enter the number "3" to "grab" an image from the screen. The computer will then prompt for the number of frames. Any integer entered less than "1" will place the ITEX boards

in a continuous "grab" mode (30 frames a second). This will provide a "real time" view of the image through the ITEX boards from the camera. Enter "0" to return back to the Main Menu.

The next step is to place the circuit to be programmed on the X-Y translation stage beneath the objective used for programming. Be sure the same objective is used since the software cannot account for changes in scale. Place the image to be used as the template in the center of the field of view of the monitor. Carefully ensure the image is correctly oriented with the movement of the X-Y translation stage. Once the image is carefully focused *on the monitor*, enter the number "4" from the Main Menu.

This is the image capturing menu once again. This time the objective is to grab a single frame of the displayed image. Enter the number "3" and then the number "0" to freeze the image on the screen. Enter "0" once again to return back to the Main Menu.

The next step is to save this image in the same directory along with the automation software. Enter the number "6" to enter the save and read menu. Enter "2" to save the image. The computer will prompt you for the starting X- and Y-coordinates. Enter "0" for both. The next prompt is for the X- and Y- size of the image. Enter "512" for both. The next prompt is for the filename of image array. The filename is your choice; however, this name must be defined appropriately in the LPROM_PARAM.h file. The final prompt is for a comment. This too is your choice; however, the computer will only record your comment up to the first space or return, which ever is first. The computer will now save the image. Upon completion the computer prompts you to "PRESS ANY KEY TO CONTINUE". After depressing any key, enter "0" to return back to the Main Menu.

The center coordinates and size of the portion of the image which is to become the actual template needs to be determined. The optimum size for the template was experimentally determined to be 30 x 30 pixels on a "squished" image. The following instruction reflect this calculation.

The first step in determining the center coordinates is to "squish" the displayed image. The pattern recognition software performs its own "squish" routine on the scanned image; thus, to obtain the correct coordinates for the template, it too must be squished.

Entering "10" will take you to the image processing menu which includes the TEST100 version of "squish". Enter "1" to begin the "squish" routine. The computer prompts you for the starting coordinates. Enter "0" for both the X and Y values. Upon completion, enter "0" to return back to the Main Menu.

Next, use the on-screen graphics aid you in determining the center coordinates. Enter "11" to enter the on-screen graphics menu. Use the circle graphic to pinpoint the actual template. Enter "3" to begin drawing the circle. The computer prompts you for the coordinates of the center of the circle. Enter your best estimations. The next prompt is for the radius. Enter "15" (the template size is 30 x 30). The circle's aspect ratio is then prompted. Enter "1" for both the X and Y directions. Finally the computer prompts you for the color of the circle. Entering a value of "0" will result in a black circle. A value of "250" results in a white circle. Repeat these graphics procedures until the circle is centered on the image to be used as the template. Record these coordinates as well as the template size, if it differs from the default value of 30 x 30, in the LPROM_PARAM.h file along with the filename of the saved image array. Enter "0" to return back the Main Menu.

Repeat these steps as necessary to generate all the templates.

Appendix D. *Programming Station Laser Power-Up and Alignment*

D.1 Introduction

This appendix provides a description of how to align and adjust the Florod optics box with the Spectra Physics 164 laser. These directions should hold for other continuous wave lasers as well. Seek the advise of a professional if there are any additional questions.

D.2 Safety

A laser can burn a retina in a fraction of a second and skin nearly as fast. Wear safety goggles designed to filter the laser energy whenever possible. ALWAYS be conscious of the distribution and direction of the laser energy. Even the best dielectric mirrors pass a small percentage of the incident laser energy. Remove all watches, rings and other jewelry which might cause unintentional incidental reflections of the beam. Be sure to illuminate the laser warning signs prior to laser power up.

D.3 Laser Power Up

Three steps are required prior to turning on the laser. The filtered, potable water must be turned on for the exciter's cooling pump. The cooling pump itself must then be turned on. Finally, the circuit breaker for the exciter must be set. When the bulb next to the "start" button illuminates, the laser is ready to be turned on. Power is adjusted through the current knob. If a buzzing sound is heard eminating from the exciter, adjust the field setting. This is the far right knob on the Model 164 control box.

D.4 Phase Alignment and Beam Conditioning

Initial set up of the phase alignment mirrors should be performed by personnel experienced in this art. All mirrors and lenses should be secured to the floatation table after positioning.

Once a beam is produced and radiates from the front phase alignment mirror, insert the beam convergence lens into the beam path. Adjust the height of the lens to ensure the beam passes through the center.

Adjust the second focusing lens to the same height. Place this lens in the beam path and carefully adjust the distance between these two lenses until a level beam emerges from the second lens.

D.5 Laser Beam Alignment

Now that the beam is properly conditioned, the remaining instructions describe how to effectively guide the beam to and out the aperture. Be sure to use dielectric mirrors and lenses wherever possible to reduce power loss. Note the directivity of the mirrors and lenses. The coatings can be applied to optimize the optics for different angles of incidence, typically 45° and 180°.

The first goal is to steer the beam squarely into the laser port on the right side of the optics box. For this thesis effort, this required both a 180° change of direction and a 3.5" reduction in elevation.

Two dielectric mirrors placed 45° to the beam and 90° to each other accomplished the change in direction. Two additional mirrors used in a periscopic fashion accomplished the change in elevation. If a beam periscope is used, be sure the mirrors are of the right type and are high quality.

Place a dielectric front surface coated mirror over the entrance to the laser port to check for proper alignment into the optics box. The beam should reflect directly back upon itself and be visible on the last dielectric mirror as a series of phase distortions. The last two mirrors and the optics box itself can be adjusted, moved or rotated to square the beam.

Once this is accomplished, remove the side access panels from the optics box. The mirrors inside the box are number M1, M2, and M3 in the order that the laser beam reaches them.

If the LRY (telescope and iris) optics, located directly above M1, are in place, carefully remove the two lenses from the top and bottom holding features. They can be replaced after the laser is aligned. If the LFA (X-Y rectangular aperture) optics are in place, open the X- and Y- apertures as wide as possible, approximately 400 on a properly zeroed digidial and remove the beam collimater from the dovetail rail between M2 and M3. Once the set screw is loosened, it can be slid off the back end of the dovetail rail. Be careful not to touch M2.

Two plexiglass alignment tools are provided in the Florod tool box. The thick one is used to align the beam between M1 and M2 and the thin one is used between M2 and M3. These tools are designed to straddle the dovetail rails and be pressed against the back panel when properly positioned. If the LFA optics are in place, the goal is to have the beam pass the through the alignment hole in the alignment tool when it is positioned just above M1 and just before it enters the top chamber. If this is not possible by adjusting M1 with the three spring loaded mounting screws, it may be necessary to reposition the optics box and have the beam reenter the port in a different location. Regardless of where in the 3/4" opening the beam enters, it must do so squarely or proper internal alignment in the optics box may not translate into maximum transmission of energy into and through the objective.

If the LRY optics are in place, the alignment tool cannot be positioned at the top of the lower chamber. Instead the iris on the LRY fixture can be constricted to serve the same purpose. Viewing the beam passing through the iris can be facilitated by the careful use of a dental mirror.

This procedure is repeated using the other alignment tool to align the beam between M2 and M3.

To make the final alignment of M3, first examine the rotary objective holder. One of the four mounting holes does not have any microadjustment screws. Insert the objective alignment tool with the black Nikon cap into this port. There is a small hole in the center of the black cap. The goal is to use the small screw adjustments on the left and right sides of the upper chamber of the optics box to adjust M3 and steer the beam through this hole.

The white material painted on the inside of the black cap is non-reflective; however, the laser light is bright and the hole, difficult to see.

Replace the objective alignment tool with the appropriate objective and adjust the remaining objective ports to focus on the same point.

D.6 Shut Down

Shut down is accomplished in the reverse order of power up. Trip the exciter circuit breaker, turn the cooling pump off and turn off the filtered potable water.

D.7 Operating Tips

Use the power meter frequently to check for changes is laser power. The expansion and contraction of the metal table and the holding fixtures causes the phase alignment mirrors to shift.

The ultraviolet light generated by the laser causes film build up on the optics. Peak power can only be maintained with clean optics. Use only denatured alcohol or approved lens and mirror cleaner.

If the cooling pump suddenly shuts down on its own, it usually indicates a dirty or clogged water filter. Replacing this filter should bring the water pressure back up and the pump will automatically turn itself back on. The exciter circuit breaker, however, will have to be reset and the start button depressed.

Appendix E. *Stepper Motor Error Charaterization Data*

E.1 Introduction

This appendix contains the data collected as part of the stepper motor error characterization effort. The data plotted on these graphs was gathered by observing and measuring the position of each memory cell, relative to a constant reference point, as it stepped across the screen. Interpolation error as a result of reading this data off the screen is +/- 0.33 μm.

This data is only for the horizontal stepping error. If the circuit is aligned correctly on the X-Y translation stage, pattern recognition performed at the beginning of each row obviates error correction in the vertical direction.

E.2 Plot Descriptions

Plots 1, 2, and 3 are the left, center, and right side measurements for motor #1 respectively. Plots 4, 5, and 6 are the left, center, and right side measurements for motor #2 respectively. The micrometer is near fully retracted on the left side measurements and near fully extended on the right side measurements. Plot #7 is data obtained by manually rotating the micrometer in 10.25 mil (approximately 26 μm) increments.

Plots 8 through 11 are the data from four consecutive rows used in determining the feasibility of error characterization and compensation. Plot #12 is plots 8 through 11 superimposed on one another.

All Y-axis are scaled in microns.

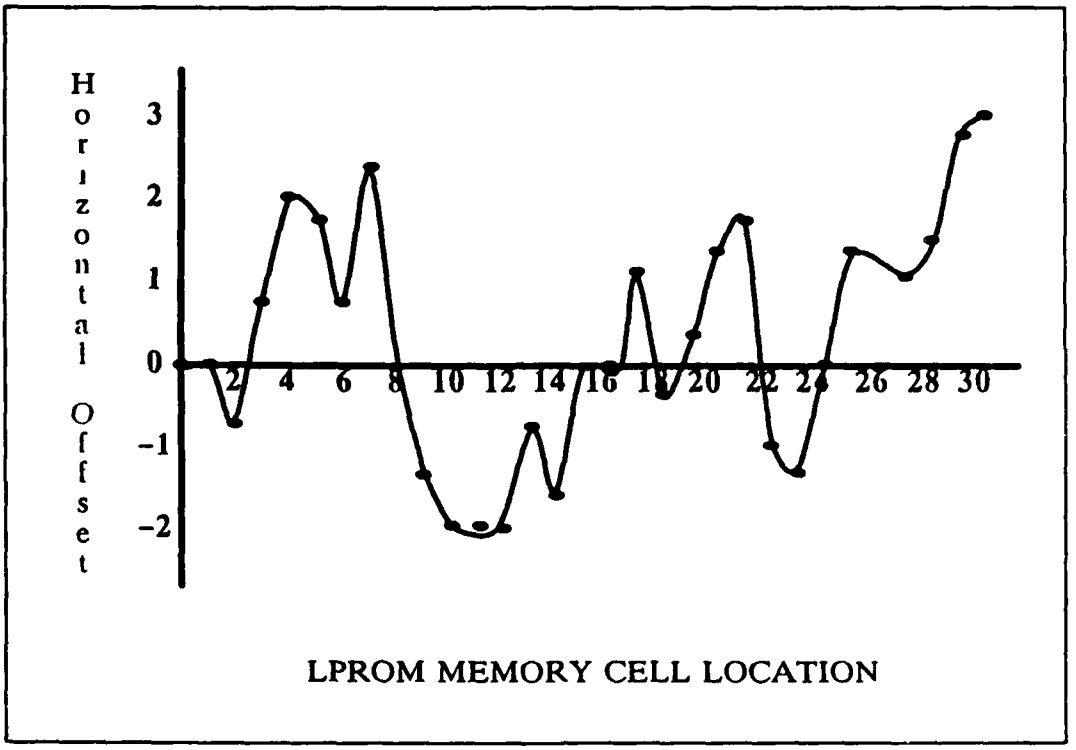


Figure E.1. Plot #1: Stepper Motor Error vs. Memory Cell Position: Left Side - Motor #1.

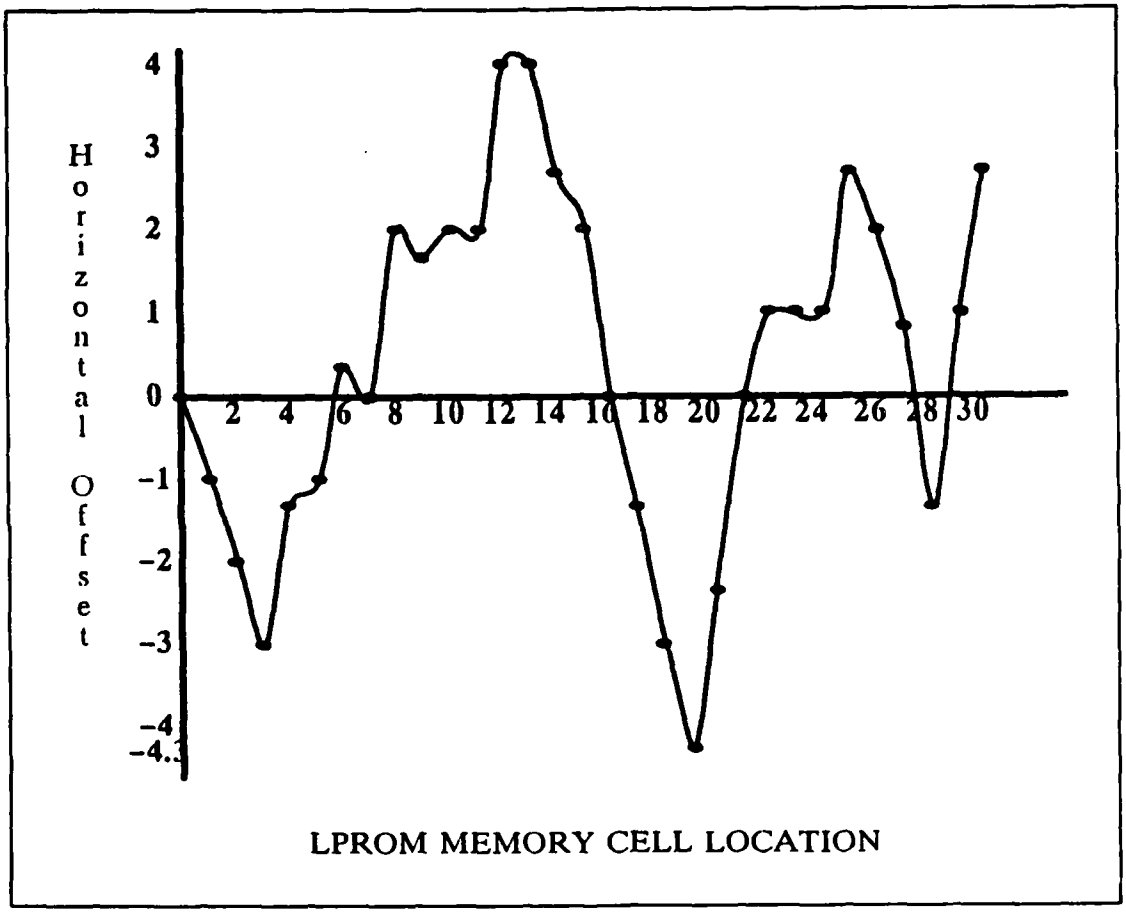


Figure E.2. Plot #2: Stepper Motor Error vs. Memory Cell Position: Center - Motor #1.

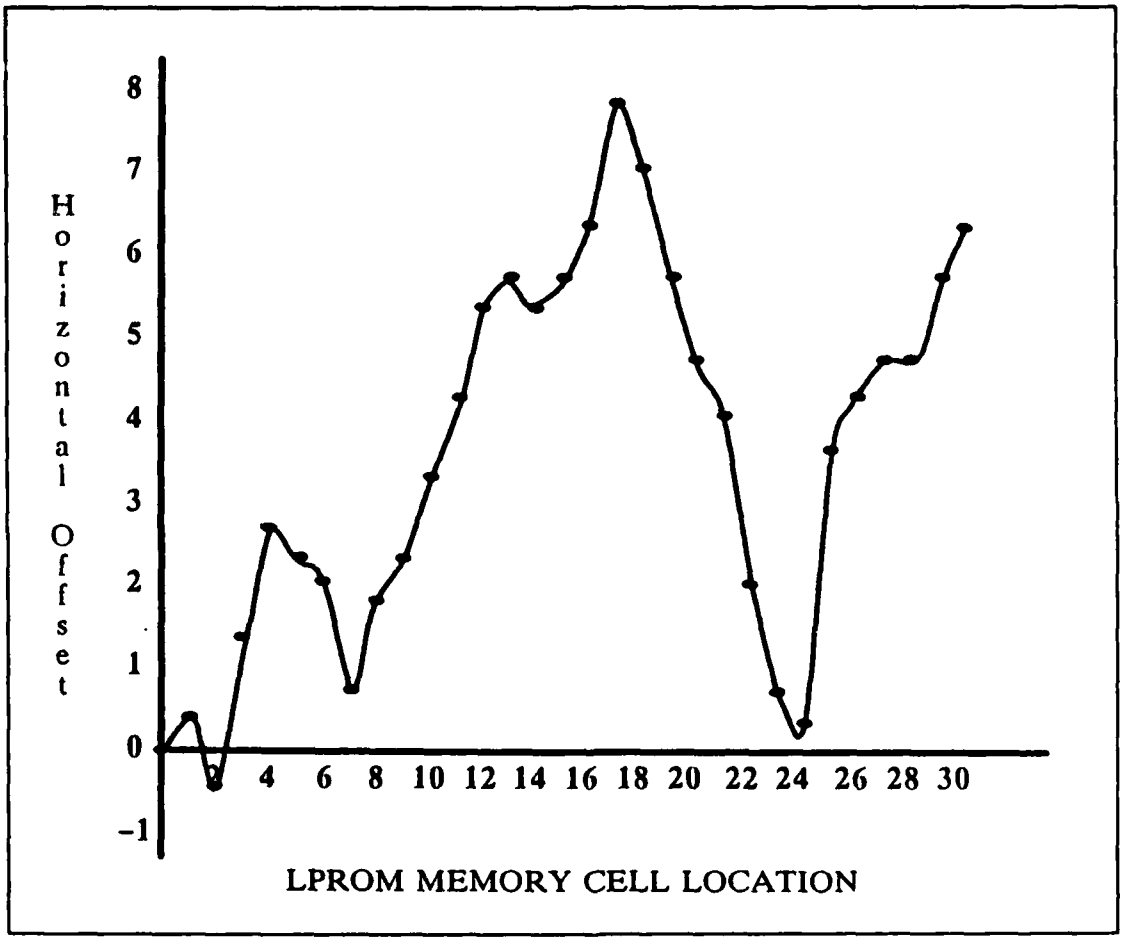


Figure E.3. Plot #3: Stepper Motor Error vs. Memory Cell Position: Right Side - Motor #1.

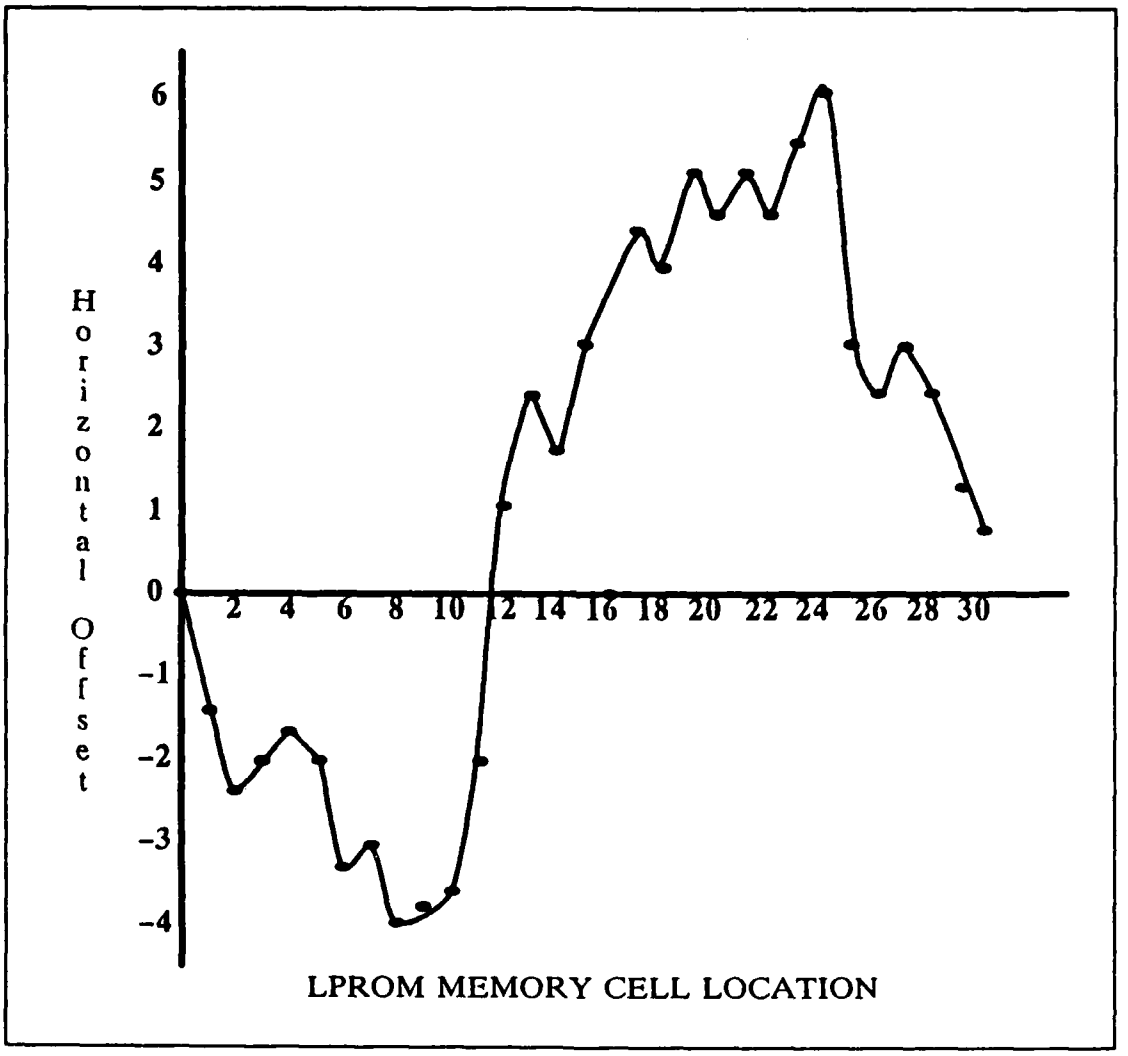


Figure E.4. Plot #4: Stepper Motor Error vs. Memory Cell Position: Left Side - Motor #2.

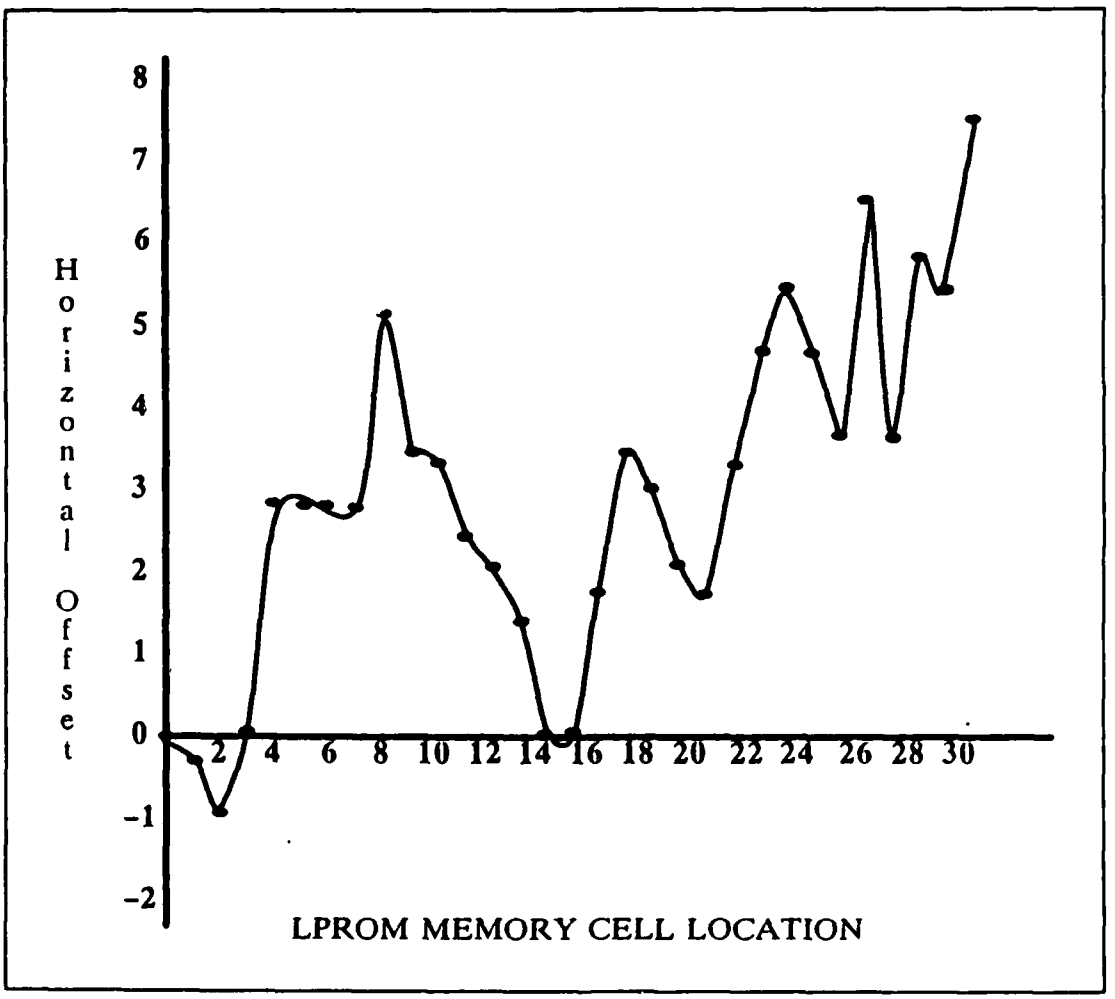


Figure E.5. Plot #5: Stepper Motor Error vs. Memory Cell Position: Center - Motor #2.

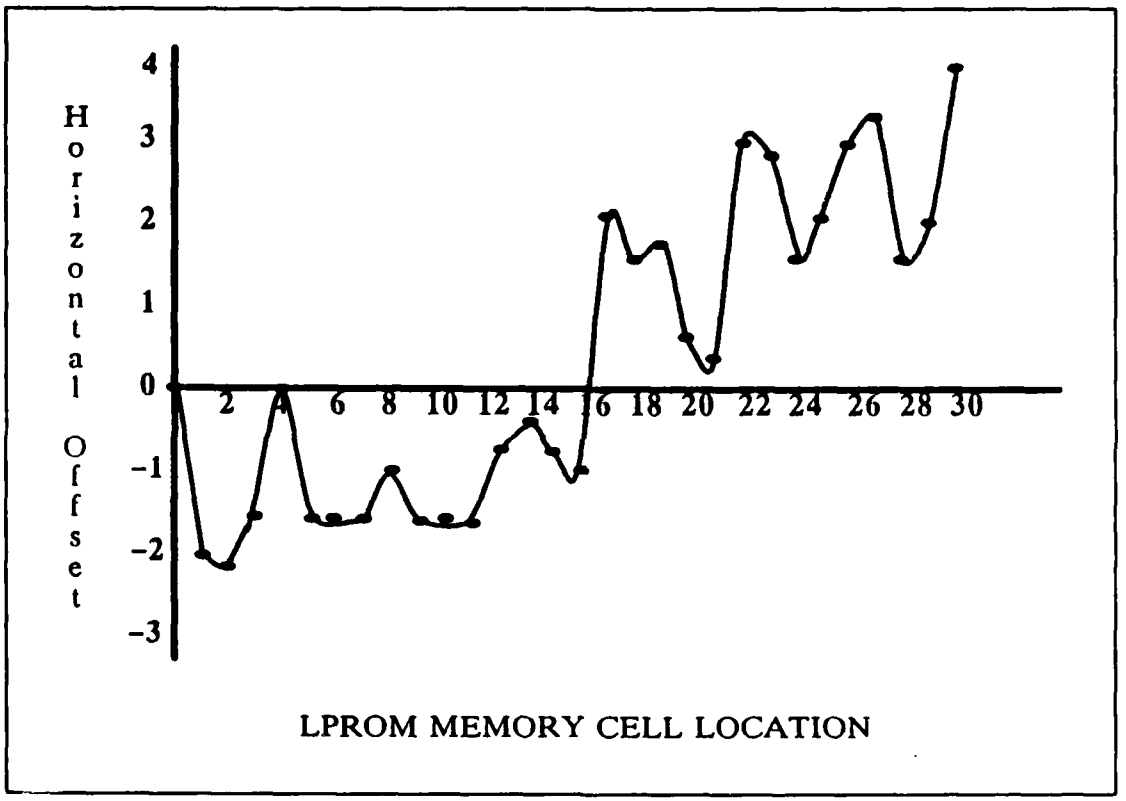


Figure E.6. Plot #6: Stepper Motor Error vs. Memory Cell Position: Right Side - Motor #2.

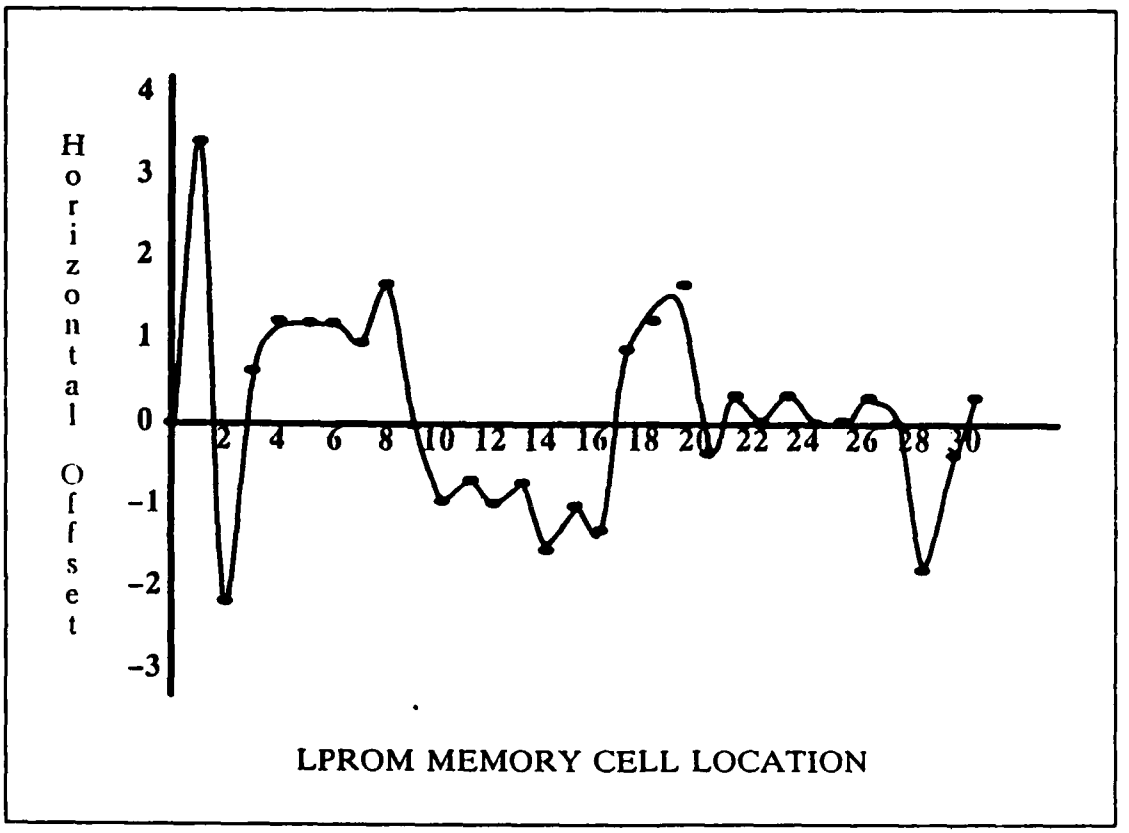


Figure E.7. Plot #7: Manual Micrometer Rotation vs. Memory Cell Position.

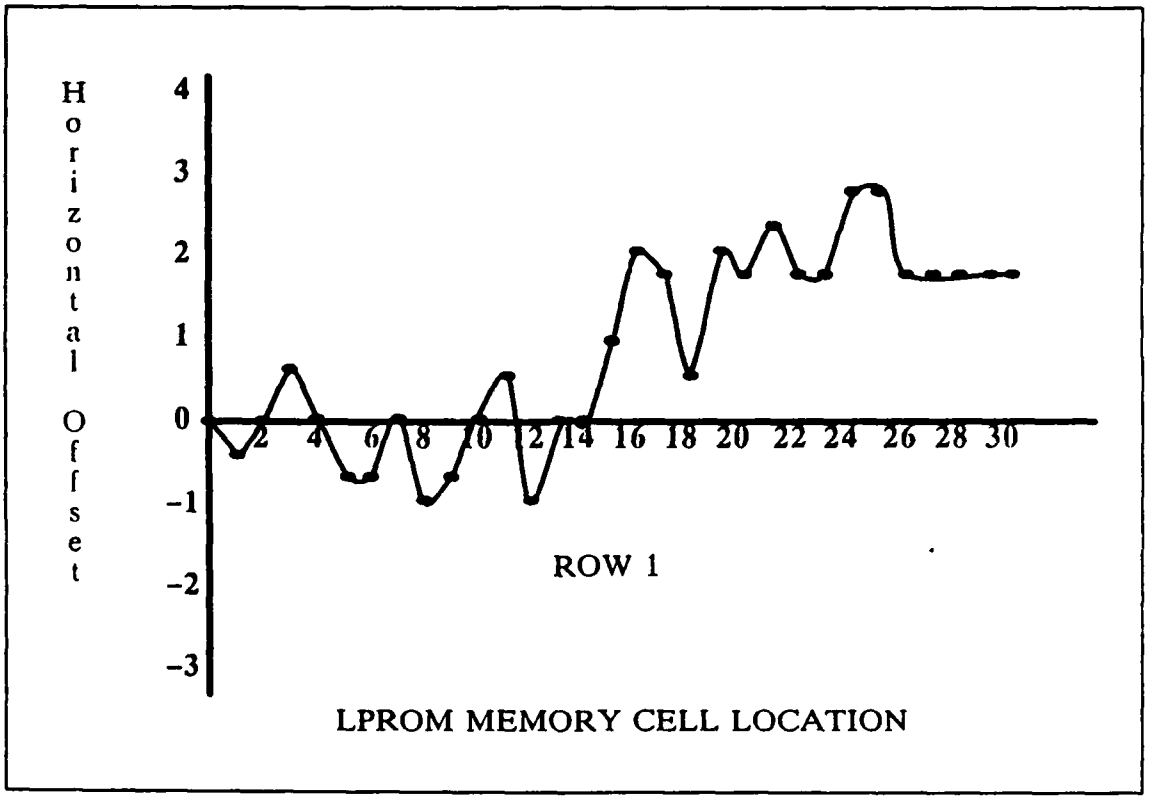


Figure E.8. Plot #8: Stepper Motor Error vs. Memory Cell Position: Row #1.

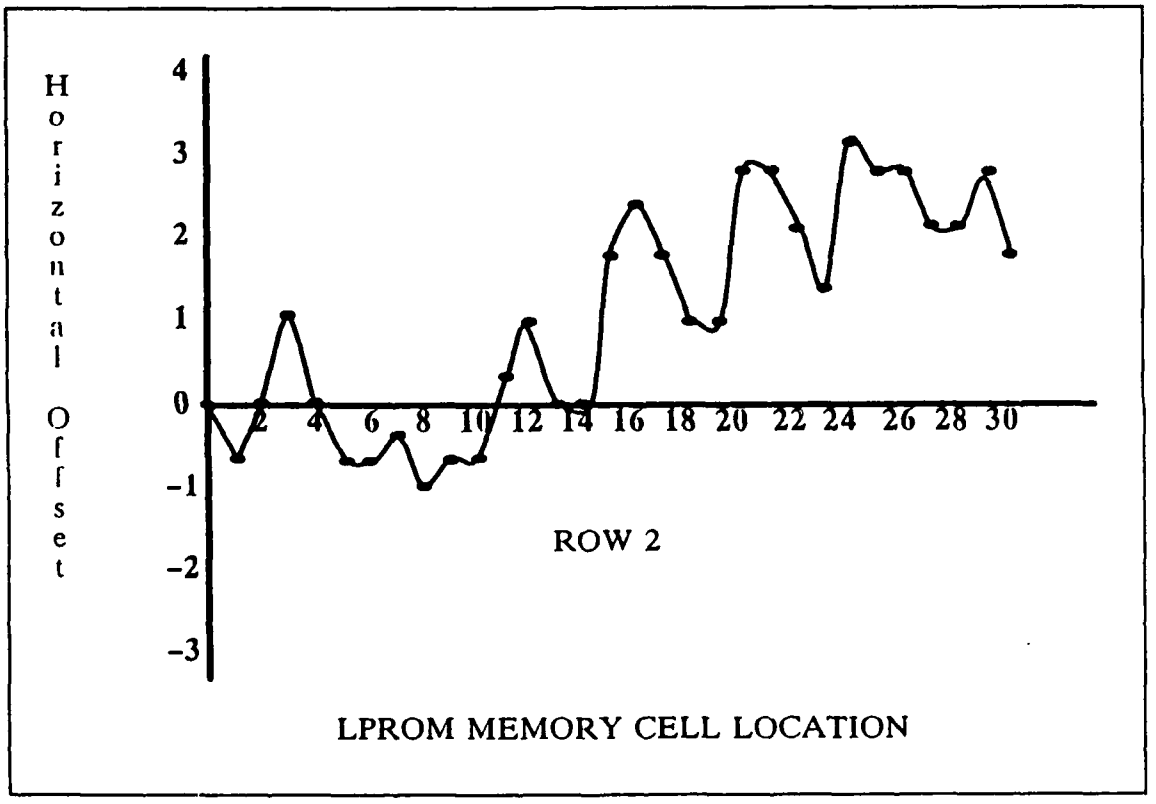


Figure E.9. Plot #9: Stepper Motor Error vs. Memory Cell Position: Row #2.

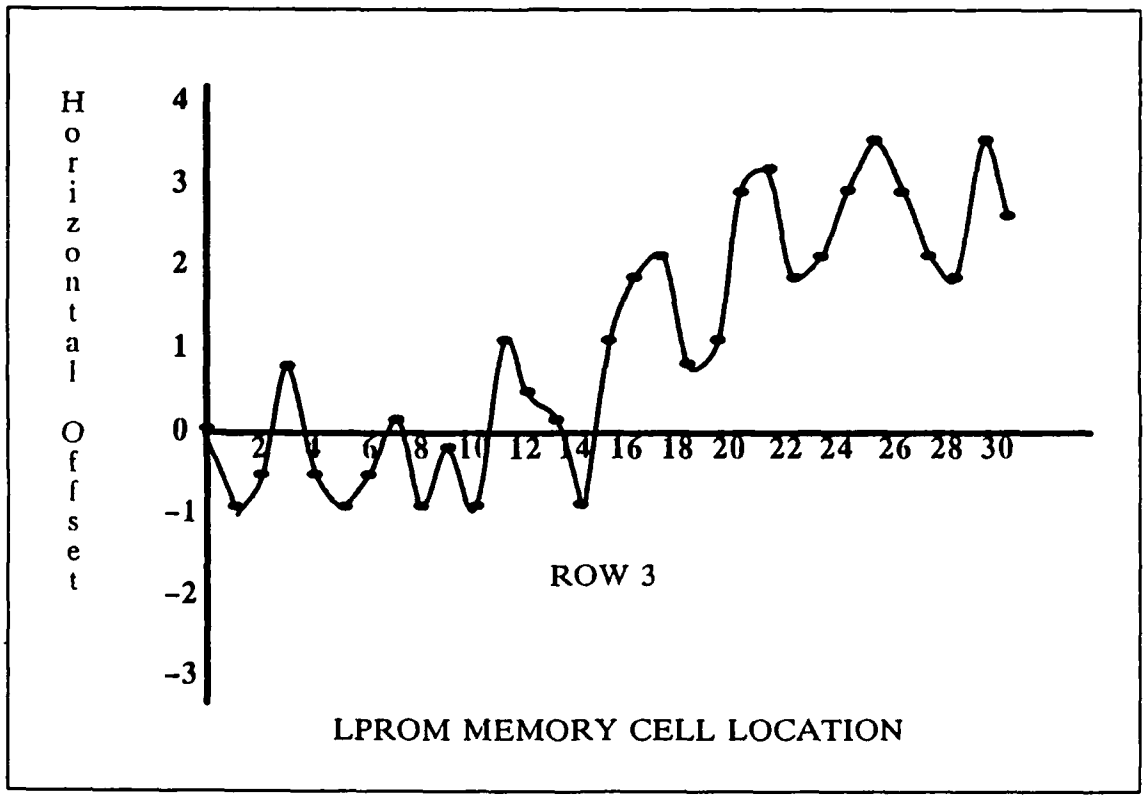


Figure E.10. Plot #10: Stepper Motor Error vs. Memory Cell Position: Row #3.

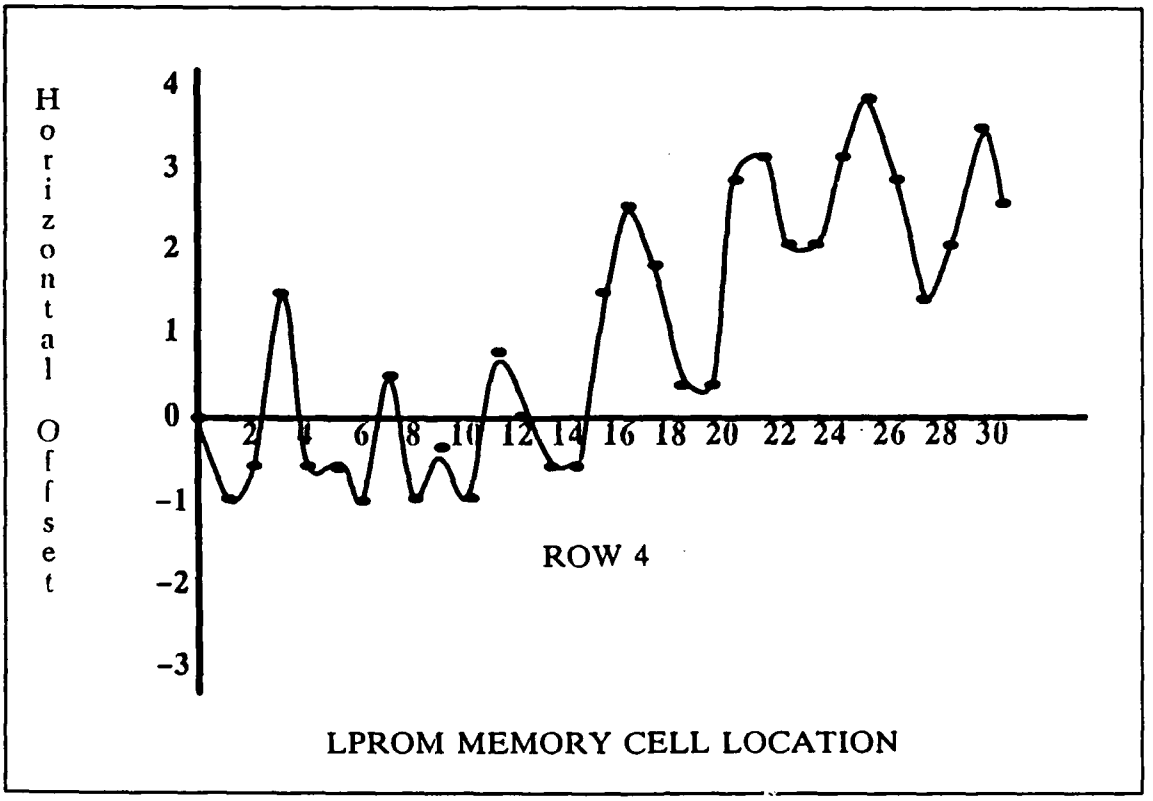


Figure E.11. Plot #11: Stepper Motor Error vs. Memory Cell Position: Row #4.

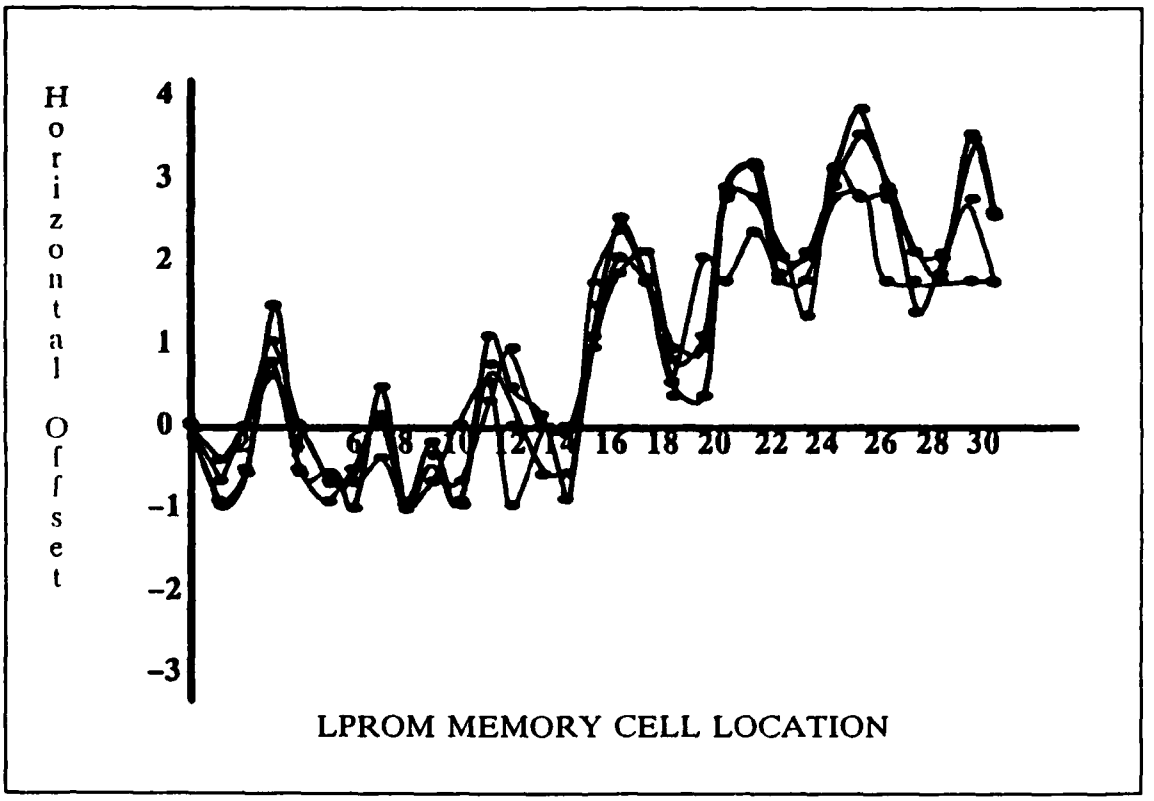


Figure E.12. Plot #12: Stepper Motor Error vs. Memory Cell Position: Rows 1-4.

Bibliography

Gla85. Glasser, L. A. and Dobberpuhl, D. W., *The Design and Analysis of VLSI Circuits*, Addison Wesley Press, Reading, Massachusetts, 1985.

Gal87. Gallagher, David M., "Rapid Prototyping of Application Specific Processors," *MS Thesis, AFIT/GE/ENG/87D-19*, School of Engineering, Air Force Institute of Technology (AU), December 1987.

Wes85. Weste, N. and Eshraghian, K., *Principles of CMOS VLSI Design*, Addison Wesley Press, Reading Massachusetts, 1985.

Fre88. Fretheim, Erik J., "Reverse Engineering VLSI Using Pattern Recognition Techniques," *MS Thesis, AFIT/GE/ENG/88J-1*, School of Engineering, Air Force Institute of Technology (AU), June 1988.

Spa87. Spanburg, Craig S., "Laser Programming Integrated Circuits," *MS Thesis, AFIT/GE/ENG/87D-62*, School of Engineering, Air Force Institute of Technology (AU), December 1987.

Ros85. Rossbach, Paul C., "Control Circuitry For High Speed VLSI Winograd Fourier Transform Processors," *MS Thesis, AFIT/GE/ENG/85D-35*, School of Engineering, Air Force Institute of Technology (AU), December 1985.

Ros86. Rossbach, Paul C., "AFIT XROM," *Custom Integrated Circuit Conference Proceedings*. December 1985.

Sze85. Sze, S. M., *Semiconductor Devices: Physics and Technology*, Wiley Press, New York, 1985.

Des81. Deshmikh, V.G.I., and others, "Pulsed Laser Modifications of SiO^2/Si Interface Properties and Minority Carrier Lifetimes," *Applied Physics Letters, Vol 39 No. 3*, 1 August 1981.

Lee87. Lee, J. J. and N. R. Strader, "CMOS ROM Arrays Programmable by Laser Beam Scanning," *IEEE Journal of Solid State Circuits, Vol SC-22, No. 4*, August 1987.

Lin85. Garverick, S L. and Edward A. Pierce, "A Single Wafer 16-Point 16-MHz FFT Processor." *IEEE Proceedings of the Custom Integrated Circuits Conference*, May 1988.

Lin86. Massachusetts Institute of Technology Lincoln Laboratory, *Lincoln Site Report Summary*, Lexington Massachusetts, 17 March, 1986.

Lin88. Massachusetts Institute of Technology Lincoln Laboratory, *Telephone Consultations with Glenn Chapman of Lincoln Laboratories*, Lexington Massachusetts, March 1988 - September 1988.

Sup87. *Installation and Operating Instructions for Modulynx Motion Controller Two Axis Type PMS085-C2A and PMS085-C2AR*. Superior Electric Company, Bristol, Connecticut, undated.

Ite87. *ITEX-100 Programmer's Manual*. Imaging Technologies Incorporated, Woburn, Massachusetts, 1986.

Yam85. Yamaguchi, Hiroshi and others. "Laser Cutting of Aluminum Stripes for Debugging Integrated Circuits," *IEEE Journal of Solid State Circuits, Vol. SC-20 No. 6*, December 1985.

Flo86. *Xenon Laser Cutter.* Product Brochure 86-1. Florod Corporation, Gardena, California, 1987.

Vin87. *UNIBLITZ Model SD-10 Shutter Timer/Driver.* Users Manual. Vincent Associates, Rochester, New York, 1987.

Mos88. *MOSIS Design Rules, Version 6,* MOSIS User Manual, University of Southern California, 1988.

Vita

Captain John J. Tillie [redacted] He entered the United States Air Force Academy on 25 June 1979. He graduated and was commissioned a Second Lieutenant on 1 June 1983. He was assigned to the 475th Test Squadron, which later became the 4484th Test Squadron, from 4 July 1983 until 8 May 1987. Captain Tillie will begin an assignment to the Air Force Systems Command Space Division at Los Angeles AFB upon completion of the Air Force Reliability and Maintainability program.

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/88D-56	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Defense Advanced Research Projects Agency; 8b. OFFICE SYMBOL (If applicable): DARPA/ISTO	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code): Washington, DC 20332	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification): A Diffusion Link Read Only Memory and Automated Programming Station (UNCLASSIFIED)

12. PERSONAL AUTHOR(S): John J. Tillie, B.S.E.E., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1988 December	141

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Laser Programming, Integrated Circuits, VHSIC, VLSI
12	06		
09	01		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Chairman: Richard W. Linderman, PhD., Captain, USAF
Assistant Professor of Electrical Engineering

12 Jan 1989

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Joseph DeGroat, Major, USAF	22b. TELEPHONE (Include Area Code): 513-255-3576 / 22c. OFFICE SYMBOL: AFIT/ENG

DD Form 1473, JUN 86 *Previous editions are obsolete.*

Abstract

Laser programmable read only memory circuits expand the applications of VLSI circuits. This thesis considers a laser programmable diffusion link as vehicle for implementing a high speed static memory circuit.

This research effort involves designing a laser programmable ROM and designing, assembling, and demonstrating an automated laser programming station. This system includes a laser and optics, a stepper motor controlled X-Y translation stage, and a camera with pattern recognition firmware.

The capabilities and limitations of this automated system are examined. The layout of the LPROM circuit illustrates design considerations for integrating a laser programmable diffusion link into a VLSI circuit. The required accuracy of the programming station is obtained at the expense of the time necessary to execute the pattern recognition firmware.